



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

## ZAVEDENÍ AGILNÍCH METOD V PRAXI

PRACTICAL APPLICATION OF AGILE METHODS

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Filip Palúch

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lenka Smolíková, Ph.D.

BRNO 2019

# Zadání diplomové práce

Ústav: Ústav informatiky  
Student: **Bc. Filip Palúch**  
Studijní program: Systémové inženýrství a informatika  
Studijní obor: Informační management  
Vedoucí práce: **Ing. Lenka Smolíková, Ph.D.**  
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

## Zavedení agilních metod v praxi

### Charakteristika problematiky úkolu:

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza současného stavu  
Návrh řešení a přínos návrhů řešení  
Závěr  
Seznam použité literatury  
Přílohy

### Cíle, kterých má být dosaženo:

Cílem diplomové práce je návrh změn a zefektivnění práce ve vývoji a vývojových týmech pomocí zavedené agilní metodiky SCRUM do vybrané firmy.

### Základní literární prameny:

DOLANSKÝ, V., a kol. Projektový management. Praha: Grada Publishing, 1996. 372 s. ISBN 80-7169-287-5.

MYSLÍN, J. Scrum: průvodce agilním vývojem softwaru. Brno: Computer Press, 2016. 167 stran. ISBN 978-80-251-4650-7.

NĚMEC, V. Projektový management. Praha: Grada, 2002. 182 s. ISBN 80-247-0392-0.

SVOZILOVÁ, A. Projektový management. 2. vyd. Praha: Grada, 2011. 392 s. ISBN 978-80-247-3-11-2.

ŠOCHOVÁ, Z. a E. KUNCE. Agilní metody řízení projektů. Brno: Computer Press, 2014. 175 s. ISBN 978-80-251-4194-6.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

---

doc. RNDr. Bedřich Půža, CSc.  
ředitel

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
děkan

## **Abstrakt**

Táto diplomová práca sa zaoberá analýzou súčasnej situácie vo firme a návrhom zmeny riadenia projektov, ktorá by tento stav vylepšila. Zmena sa orientuje na nasadenie agilnej metodiky Scrum, ktorej teoretické základy sú vysvetlené v rámci prvej kapitoly. Na základe zistených poznatkov z analytickej časti sú navrhnuté konkrétne zmeny založené na princípoch Scrumu, ktoré sú popísané v jednotlivých podkapitolách návrhovej časti.

## **Abstract**

This diploma thesis is focused on current situation analysis of the company with particular designed change of project management. This change is aimed to set up a Scrum agile method. Basics of Scrum agile method are explained in the first chapter of this diploma thesis. Based on the knowledge gained from analytical chapter of this thesis there are designed specific changes according Scrum principles in the last chapter.

## **Kľúčové slová**

agilné metódy, scrum, projektové riadenie

## **Keywords**

agile methods, scrum, project management

### **Bibliografická citácia**

PALÚCH, Filip. *Zavedení agilních metod v praxi* [online]. Brno, 2019 [cit. 2019-05-08]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119608>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Lenka Smolíková.

### **Čestné prehlásenie**

Prehlasujem, že predložená diplomová práca je pôvodná a spracoval som ju samostatne.  
Prehlasujem, že citácia použitých prameňov je úplná, že som v práci neporušil autorské práva (v zmysle Zákona č. 121/2000 Sb., o práve autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 9. mája 2019

.....

Filip Palúch

### **Pod'akovanie**

Chcel by som sa poďakovať vedúcej mojej diplomovej práce Ing. Lenke Smolíkovej, Ph.D. za odborné vedenie, dohľad a konzultácie, ktoré mi v priebehu písania tejto diplomovej práce poskytla, a tak isto mojim rodičom, ktorí ma počas písania práce podporovali.

# OBSAH

ÚVOD .....	12
CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA .....	14
1 TEORETICKÉ VÝCHODISKÁ PRÁCE .....	16
1.1 Projektový management.....	16
1.2 Projekt .....	17
1.3 Projektový trojimperatív .....	19
1.4 Životný cyklus projektu .....	20
1.4.1 Predprojektová fáza .....	20
1.4.2 Projektová fáza .....	24
1.4.3 Poprojektová fáza .....	24
1.5 Produkt Projektu.....	24
1.6 Časová analýza projektu .....	25
1.6.1 Metóda kritickej cesty CPM .....	26
1.6.2 Metóda PERT .....	26
1.7 Riadenie rizík .....	28
1.7.1 Riziko.....	28
1.7.2 Analýza rizika .....	29
1.7.3 Skórovacia metóda.....	31
1.8 Tradičné metodiky vývoja projektu (softwaru).....	32
1.8.1 Software .....	33
1.8.2 Vodopádový model (Waterfall model) .....	33
1.8.3 Iteratívny model.....	34
1.8.4 Inkrementálny model .....	35
1.8.5 Špirálový model.....	36



1.9	Agilné metodiky .....	36
1.9.1	Čo znamená agilne? .....	37
1.9.2	Agilný manifest.....	37
1.9.3	Dôvody prechodu na agile .....	38
1.9.4	Typy agilných metodík .....	39
1.9.5	Lean .....	40
1.9.6	Extrémne programovanie.....	41
1.9.7	Kanban .....	42
1.9.8	Scrum .....	43
2	ANALÝZA SÚČASNEJ SITUÁCIE .....	57
2.1	Predstavenie spoločnosti .....	57
2.1.1	Zameranie spoločnosti .....	58
2.1.2	Smer rozvoja spoločnosti v budúcnosti .....	59
2.2	Analýza interných faktorov (Rozbor 7S faktorov).....	59
2.3	Analýza okolitého prostredia organizácie (SLEPT analýza) .....	63
2.4	Porterova analýza piatich síl .....	64
2.5	Analýza priebehu vývoja vo firme .....	66
2.6	SWOT analýza .....	69
3	NÁVRH RIEŠENÍ A PRÍNOS NÁVRHOV RIEŠENÍ .....	72
3.1	Návrh nových rolí .....	73
3.1.1	Scrum Master .....	73
3.1.2	Product Owner .....	75
3.1.3	Product Owner Proxy.....	77
3.1.4	Scrum Team Member .....	78
3.1.5	Zákazník.....	79
3.1.6	Scrum Team .....	80

3.2	Návrh meetingov a základných artefaktov Scrumu .....	81
3.2.1	Backlog .....	81
3.2.2	Sprint.....	85
3.2.3	Zahajovací meeting.....	86
3.2.4	Backlog Refinement Meeting .....	87
3.2.5	Grooming .....	89
3.2.6	Plánovanie.....	91
3.2.7	Scrum Meeting.....	94
3.2.8	Retrospektíva .....	95
3.2.9	Sprint Review .....	98
3.2.10	Grafický návrh procesu vývoja.....	100
3.3	Nástroje na riadenie Scrumu .....	101
3.3.1	Jira Software .....	101
3.3.2	Scrum Tabula.....	101
3.3.3	Úloha v Jira Software .....	103
3.4	Časová analýza.....	104
3.4.1	PERT analýza .....	105
3.4.2	Analýza rizík.....	109
3.5	Náklady na zavedenie zmeny vo firme .....	112
3.5.1	Náklady na zavedenie Scrumu.....	113
3.5.2	Náklady na zavedenie Jira Software .....	114
3.6	Prínos návrhov .....	117
ZÁVER .....		118
ZOZNAM POUŽITEJ LITERATÚRY .....		120
ZOZNAM TABULIEK .....		122
ZOZNAM OBRÁZKOV .....		123

ZOZNAM GRAFOV ..... 125

## ÚVOD

Prvé zmienky o pojme projektový management sa objavili už v dávnych časoch staroveku, keď boli potrebné rôzne metódy, postupy a techniky ku zvládaniu rozsiahlych a organizačne náročnejších akcií. Aj čo sa týka získavania zdrojov, ktoré sú potrebné k riadeniu projektov môžeme vidieť pomerne veľké rozdiely. V prípade nedostatku zdrojov ich jednotlivé civilizácie nadobúdali vojenskými akciami v podobe zlata a otrokov. V minulosti sa nemuseli natoľko venovať časovým aspektom akcií. Keď sa pozrieme na stavby, ktoré sa budovali desaťročia a prežili niekoľko architektov tak dnešná doba je oveľa náročnejšia na čas. A práve náročnosť na čas a zdroje nútila spoločnosť vytvoriť systém ktorý by sa lepšie prispôbil dnešnej, dynamickejšej dobe.

Postupom času sa vznikajúce spoločnosti začali prispôbovať zmenám, ktoré začali ovplyvňovať obchod. Práve rýchlosť reakcie na zmeny začala byť kľúčová. Tieto zmeny sa týkali hlavne informačných technológií, kde môžeme vidieť obrovské pokroky najmä v 20. storočí. Kedysi poslanie správy trvalo podstatne viac času, kým ju niekto zaviezol na dané miesto a potom priniesol naspäť. Dnes e-mailová komunikácia prebieha takmer okamžite. Spoločnosti, organizácie a firmy sa museli podriadiť dobe a naučiť sa agilne prispôbovať podmienkam, pokiaľ chceli byť v budúcnosti konkurencieschopné a prežiť na trhu, ktorý sa postupom času preplňal.

Vznik nových technológií a produktov, prípadne stanovenie cieľov do budúcnosti vytvorili základ pre rozvoj projektového manažmentu v spoločnosti. Prvá zmienka o projektovom manažmente sa datuje do 30. rokov 20. storočia. Výrazný podiel na tom mala hospodárska kríza a od tej doby sa projektový manažment začal využívať okrem stavebníctva a priemyslu aj na spomínané informačné technológie. Od tej doby ide vývoj projektového manažmentu ruku v ruku s rozvojom v IT či už na úrovni software, hardware alebo komplikovanejších informačných systémov.

Postupom času sa projektový manažment stal hlavnou súčasťou väčšiny podnikateľských aktivít. Do súčasnosti bolo vytvorených množstvo opatrení, vyhlášok, noriem a štandardov, ktoré sa týkajú projektového manažmentu. Podobne aj nástroje projektového manažmentu sa rozvíjali či už pri menších alebo aj rozsiahlejších projektoch.

Jedným z takýchto nástrojov, ktorý sa výrazne používa a podporuje vývoj projektov prevažne vo firmách zaoberajúcimi sa informačnými technológiami je agilný prístup. Práve agilné metodiky nám umožňujú reagovať na zmeny rýchlejšie a pružnejšie, čo v dnešnej dobe tak, ako sme už spomínali, hrá veľkú rolu. Podstatnú úlohu v tomto prístupe zohráva zákazník, ktorý je začlenený do vývoja a konzultuje priebežný stav či sa projekt vyvíja smerom akým by sa mal. Pri využívaní agile zohráva podstatnú časť komunikácia jednak medzi členmi tímu, ktorý sa na projekte podieľajú a tak isto aj zákazníkom.

Úlohou tejto diplomovej práce bude zmeniť aktuálny stav firmy s informačným systémom pomocou agilného prístupu, ktorý budeme zavádzať.

## **CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA**

Cieľom tejto práce je návrh a zavedenie agilnej metodiky Scrum v spoločnosti iXperta s.r.o., ktorá by mohla prispieť k zlepšeniu a zefektívneniu procesu vývoja v rámci vývojových projektov naprieč všetkými tímami vo firme.

Na začiatku tejto práce sú vysvetlené a zadefinované základné pojmy týkajúce sa projektového manažmentu a tak isto sú definované tradičné metódy vývoja software a ich základné typy. Následne sú popísané pojmy a základné charakteristiky týkajúce sa agilných metodík. Podstatnou časťou je vysvetlenie princípov agilného manifestu a pravidiel agilného vývoja. Sú objasnené základné výhody a nevýhody použitia agilných metodík v praxi a tak isto dôvody zavedenia agilného prístupu. Na konci kapitoly sú porovnané jednotlivé typy agilných metodík s ich popismi. V rámci tejto časti diplomovej práce by mali byť viditeľné rozdiely medzi prístupmi pri tradičných metódach vývoja a agilnom prístupe pri vývoji.

V nasledujúcej kapitole analytickej časti diplomovej práce je predstavená spoločnosť iXperta s.r.o. a tak isto aj jej hlavné zameranie. Využitím metód a analýz, ktoré sú definované v kapitole Teoretické východiská práce, ako je Analýza interných faktorov, Analýza obecného okolia, Porterova analýza a SWOT analýza je popísaná situácia firmy na trhu a tak isto aj jej hlavné problémy, ktoré by mala odstrániť a prípadne urobiť potrebné zmeny. V rámci analýzy je detailne popísaná metóda priebehu vývoja projektov s jednotlivými fázami, ktorá sa najviac stotožňuje s metódou vodopádového modelu. Tu sú spomenuté problémy, ktoré výrazne ovplyvňujú výsledný produkt, a vývojové tímy ich nie sú schopné efektívne vyriešiť alebo im predchádzať. V tejto kapitole sa poukazuje na výrazné nevýhody doterajšieho vývoja a nutnosť zavedenia agilného prístupu.

Posledná kapitola sa venuje návrhovej časti diplomovej práce, kde sú použité konkrétne metódy projektového manažmentu. Na začiatku tejto kapitoly sú vysvetlené základné pojmy a artefakty agilnej metodiky Scrum. Na základe výstupov z predošlej kapitoly týkajúcej sa analýzy sú definované činnosti, ktoré sú potrebné na zavedenie metodiky Scrum do spoločnosti. V jednotlivých podkapitolách je popísané nasadenie Scrumu do vývojových tímov a tak isto všetky podporné nástroje, ktoré pomáhajú k lepšiemu

riadeniu. Na zobrazenie činností potrebných k zavedeniu zmeny je použitá metóda PERT analýzy, ktorá uvádza priebeh činností celej transformácie s konkrétnymi údajmi. Na identifikovanie a vymedzenie rizík, ktoré môžu nastať pri tejto zmene bude použitá Skórovacia metóda.

# 1 TEORETICKÉ VÝCHODISKÁ PRÁCE

Táto kapitola sa bude venovať vysvetleniu a rozobratiu základných pojmov, ktoré budeme používať v ďalších častiach práce.

## 1.1 Projektový management

Hlavnou otázkou súčasného manažmentu je riadenie v podmienkach neustálych zmien. Bez ohľadu na typ organizácie, musí sa neustále prispôbovať zmenám, reagovať na podnety dynamického okolia. Preto môžeme povedať, že reagovanie na podnety sa stáva kľúčovým faktorom úspechu. Projektový manažment si môžeme definovať pomocou týchto dvoch definícií:

Projektový manažment je súhrn aktivít spočívajúci hlavne v plánovaní, organizovaní, riadení a kontrole zdrojov spoločnosti s relatívne krátkodobým cieľom, ktorý bol stanovený pre realizáciu špecifických cieľov a zámerov.[1]

Projektový manažment je aplikácia znalostí, schopností, nástrojov a technológií na aktivity projektu tak aby splnili požiadavky projektu.[2]

Obe definície sú vo svojom znení odlišné, ale ich podstata je obdobná: projekt je určité krátkodobo vynaložené úsilie doprevádzané aplikáciou znalostí a metód, ktorých účelom je premena materiálnych a nemateriálnych zdrojov na súbor predmetov, služieb alebo ich kombinácie tak, aby boli dosiahnuté vytýčené ciele.[3]

Medzi základných päť elementov projektového managementu patrí:

- Projektová komunikácia
- Tímová spolupráca
- Životný cyklus projektu
- Vlastné súčasti projektového managementu
- Organizačný záväzok [3]

Tieto základné prvky sú súčasťou každého projektového manažmentu. Je potrebné ešte spomenúť rozdiel medzi pojmami obecný manažment a projektový manažment.



Obecný manažment a projektový management vychádzajú z rovnakých princípov, metód, zásad a nástrojov riadenia, vykazujú však odlišnosti vo vybraných nástrojoch, metódach, analytických technikách, ktoré sú prevažne určené pre projektový manažment. Projektový manažment sa líši od všeobecného manažmentu používaného v líniovo riadenej organizácii hlavne svojou dočasnou a prídelením zdrojov pre jeho realizáciu podľa potrieb projektu.[3]

Je potrebné spomenúť, že projektový manažment má svoje výhody, ale na druhej strane má aj svoje nevýhody. Dôvody využitia projektového manažmentu sú:

- Jednoznačne určené role a zodpovednosti
- Jednoznačne identifikovaný časový a nákladový rámec
- Realizačné zdroje prídelené po dobu trvania projektu
- Sledovanie skutočného priebehu oproti plánu, možnosť korekcie v prípade odchýlky
- Systém rozdelenia zodpovednosti za riadenie projektu
- Princípy riadenia
- Systémový prístup k riadeniu projektu je výhodou znovu použiteľnosti [3]

Úspešným projektom môžeme nazvať projekt, v ktorom sú splnené plánované ciele projektu, v naplánovanom čase a pri využití a dodržaní predpokladaných nákladov.[4]

Bolo spomenuté, že projektový manažment má aj svoje problematické stránky, ktorých riešenie záleží na skúsenosti, pripravenosti a talente projektového manažéra. Medzi nevýhody patria:

- Komplexný rozsah projektov a zaradenie projektov do hierarchie
- Špecifické požiadavky zákazníka projektu
- Organizačné zmeny v priebehu projektu
- Riziká projektu a vonkajšie vplyvy
- Zmeny v technológiách
- Plánovanie a oceňovanie v predstihu [3]

## **1.2 Projekt**

Definícia pojmu projekt nemusí byť taká jednoznačná, nakoľko môže byť použitá v rôznych odvetviach a tým pádom sa môže líšiť. Napríklad práca architekta môže byť

označená ako projekt. Podobne aj v stavebníctve a strojárstve je projekt často používaným výrazom. Tak by sme mohli pokračovať aj do ďalších odborov kde by sa výskyt tohto slova našiel. Tieto použitia majú spoločné to, že sú ekvivalentné s označením návrh.

Pod návrhom si môžeme predstaviť:

- Špecifikáciu funkčných parametrov
- Technické riešenie
- Výber použitej technológie
- Technickú dokumentáciu[5]

Projekt je najdôležitejším prvkom projektového riadenia. Ako už bolo spomenuté, jeho definície sa môžu líšiť. Skúsme si definovať projekt pomocou týchto dvoch definícií.

Projekt je jedinečný sled aktivít a úloh, ktorý má:

- Daný špecifický cieľ, ktorý má byť jeho realizáciou splnený
- Dátum začiatku a konca uskutočnenia
- Stanovený rámec pre čerpanie zdrojov potrebných pre jeho realizáciu[1]

Projekt je dočasné úsilie vynaložené na vytvorenie unikátneho produktu, služby alebo určitého výsledku.[2]

Z definícií, ktoré boli uvedené môžeme povedať, že návrh je súčasťou projektu od začiatku, alebo je vytvorený až v priebehu projektu. Napriek tomu, že sme si definovali pojem projekt, ako vlastne zistíme, že sa daná akcia považuje za projekt? O tom nám povedia atribúty, ktoré rozhodnú, či sa daná akcia považuje za projekt alebo nie. Týmito atribútmi sú:

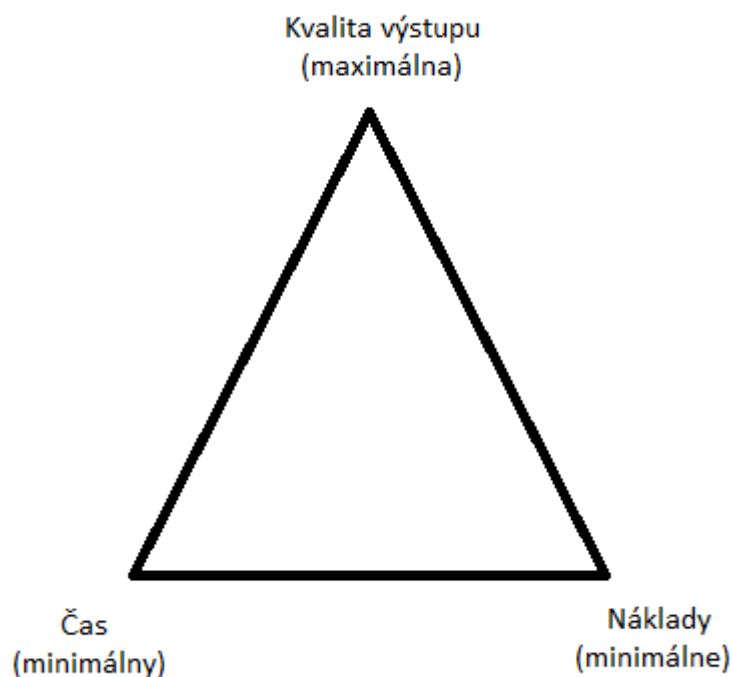
- Jedinečnosť – je určená výsledkom projektu. Musíme presne vedieť, čo má byť výsledkom celého snaženia, ako spoznáme, že sme došli do cieľa a každý, kto sa podieľa na projekte to musí chápať tak isto
- Vymedzenosť – je určená časom, pridelenými nákladmi a disponibilnými zdrojmi
- Realizácia projektovým tímom – dočasná organizačná štruktúra, ktorá je ukončením projektu zrušená
- Nadpriemerné riziko – je spojené hlavne s novotou riešeného problému. Je spôsobené hlavne neurčitou pri definovaní cieľov a spôsobom ich realizácie.

- Komplexnosť a zložitosť – vyplýva zo zložitosti tvorby produktu projektu a na výstupe[6]

### 1.3 Projektový trojimperatív

Stanovený cieľ vyjadruje to, čo treba urobiť, ale tak isto musíme určiť, kedy sa to má urobiť a za koľko. Takto definovanému cieľu sa v projektovom managemente hovorí trojimperatív. Definícia nám vysvetlí dôvod tohto názvu lepšie.

*„Trojimperatív popisuje vzťah medzi plánovaným cieľom, plánovaným termínom a plánovanými nákladmi. Jedná sa vlastne o cieľ projektu vyjadrený v dimenziách výsledok(kvalita), čas a náklady. Preto sa niekedy nazýva trojrozmerným cieľom.“ (7, s.51)*



**Obrázok 1: Trojimperatív projektu** (Zdroj: vlastné spracovanie podľa [7])

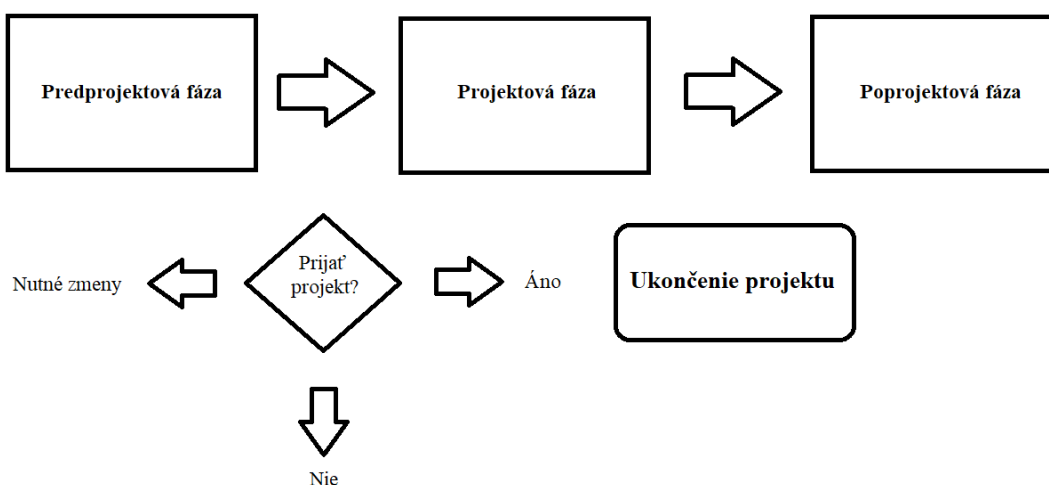
Aby bol projekt úspešne ukončený, platí, že tento dynamický systém musí byť v tomto priestore udržiavaný v rovnováhe. K tomu, aby bol tento predpoklad naplnený, slúži plán projektu, podľa ktorého je sled prác koordinovaný, s aktuálnym pôsobením kontrolných systémov, ktoré ich monitorujú, nakoľko je daný systém udržiavaný

v rámci stanovených limitov. Za ideálneho stavu a s dobre pripraveným plánom, by mali byť šance na úspešné dokončenie veľmi vysoké, ale na projekt pôsobia vplyvy vyvolávajúce zmeny a rizikové situácie, ktoré sú príčinou výkyvu systémov z jeho rovnovážneho stavu.[3]

## 1.4 Životný cyklus projektu

Čas je jedným z hlavných parametrov a kritérií úspešnosti projektu. Preto sa môžeme na projekt pozerat' ako na skupinu za sebou nasledujúcich fáz, ktoré vyjadrujú priebeh vývoja projektu. Pritom môžeme povedať, že z pohľadu projektu jedna fáza predstavuje skupinu na seba súvisiacich procesov.[7]

Aj napriek tomu, že jednotlivé projekty sa môžu medzi sebou líšiť, je možné určiť aspoň základné fázy, ktoré sa týkajú každého projektu. Teraz si tieto tri základné projektové fázy bližšie špecifikujeme.



Obrázok 2: Fázy projektu (Zdroj: vlastné spracovanie podľa [8])

### 1.4.1 Predprojektová fáza

V tejto fáze dochádza ku analýze možností realizácie projektu a zvažuje sa či je projekt životaschopný a či po ňom existuje dopyt na trhu. Pokiaľ prieskum potvrdí príležitosti pre projekt, zvažuje sa spôsob jeho realizácie. Výsledkom tejto fázy by malo byť rozhodnutie či tento projekt realizovať alebo nie.[7] Túto fázu môžeme rozdeliť do etáp:

- Identifikácia príležitosti zmien

- Predbežný výber a definovanie projektu
- Podporné alebo funkcionálne štúdie
- Štúdie vykonania

#### **1.4.1.1 SLEPT analýza**

Táto analýza slúži k analýze zmien v obyčajnom prostredí spoločnosti. Pozoruje a hodnotí jednotlivé externé faktory, ktoré by mohli projekt nejako ovplyvniť. Hodnotenie externých faktorov prebieha z piatich hľadísk:

- Sociálne
- Legislatívne a právne
- Ekonomické
- Politické
- Technické alebo Technologické [7]

Analýza nerieši len aktuálny stav spoločnosti, ale rieši aj to, ako sa môže prostredie firmy v budúcnosti vyvíjať a aké zmeny sa dajú očakávať. Výsledkom analýzy je prehľad faktorov vonkajšieho prostredia, ktoré môžu projekt ovplyvniť. [7]

#### **1.4.1.2 7S analýza**

Na základe analýzy stránok, ktoré majú pre manažéra zásadný význam je možné určiť kritické faktory, ktoré ovplyvňujú úspech firmy. Úspešná firma je ovplyvňovaná siedmimi vnútornými, vzájomne závislými faktormi, ktoré musia byť vzájomne rozvíjané. Tieto faktory definujú rámec 7S faktorov firmy McKinsey.[19]

Jednotlivé faktory 7S Modelu:

- Stratégia – obvykle vychádza z vízie firmy (predstav majiteľov firmy) a z konkrétneho poslania firmy (služby, výroba)
- Štruktúra firmy – základnou úlohou organizačnej štruktúry je rozdelenie úloh, kompetencií a právomocí medzi pracovníkov organizácie
- Spolupracovníci – ľudia sú hlavným zdrojom zvyšovania výkonnosti, tento faktor hovorí o vzťahoch, motivácii a správaní voči firme
- Schopnosti – schopnosti a skúsenosti daného kolektívu v sledovanom odbore
- Štýl riadenia – popisuje ako manažment riadi a pristupuje k problémom

- Systémy a postupy firmy – pod týmto sa chápu všetky informačné procedúry, ktoré v organizácii prebiehajú (ručné a automatizované spracovanie informácií)
- Zdieľané hodnoty – predstavujú súhrn predstáv, mýtov, prístupov členov kolektívu vo firme [14]

#### 1.4.1.3 SWOT analýza

SWOT analýza patrí medzi základné nástroje, ktorý skúma kľúčové otázky organizácie na základe podrobnej analýzy ich silných a slabých stránok a dôležitých vplyvov z vonkajšieho prostredia. Dôležité je stanovenie predmetu analýzy, jeho účelu a časového horizontu, v rámci ktorého budú diskutované príležitosti a hrozby. Analýza môže byť aplikovaná na organizácii ako takej, projekte, projektovom tíme alebo návrhu na projekt v jeho predprojektovej fáze.[7]

**Tabuľka 1: SWOT analýza** (Zdroj: vlastné spracovanie)

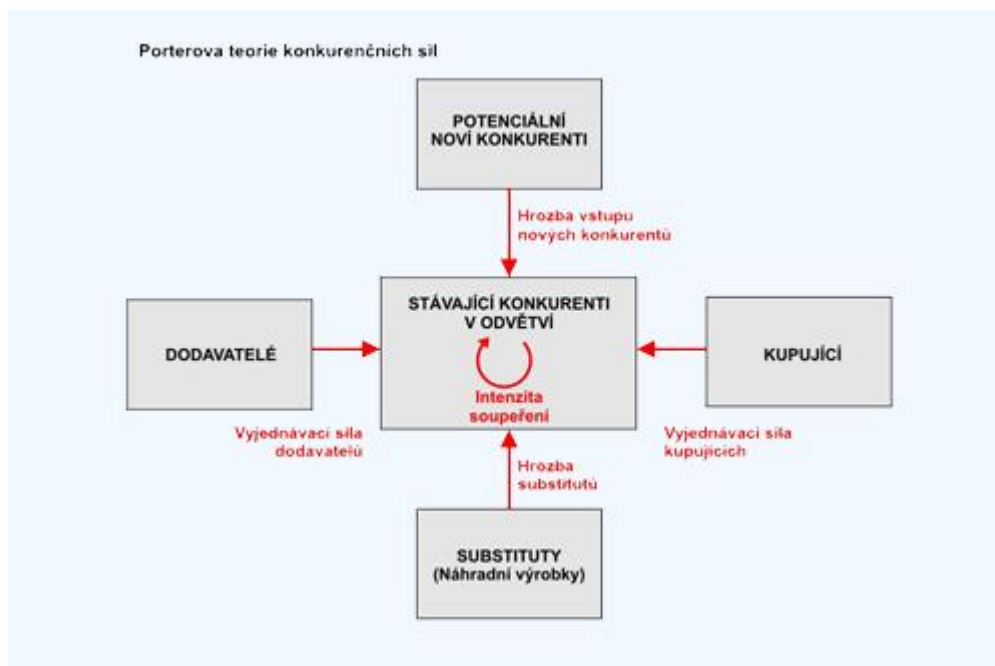
<b>Silné stránky</b>	<b>Slabé stránky</b>
<b>Príležitosti</b>	<b>Hrozby</b>

V rámci tejto analýzy sa rozlišujú štyri základné faktory, ktoré sú rozdelené do štyroch kvadrantov tak, ako je to zobrazené v tabuľke vyššie.

- Opportunities – Príležitosti – jedná sa o súhrn možných oblastí vývoja a zdokonalení, ktoré môže spoločnosť využiť vo svoj prospech.
- Threats – Hrozby – zameriava sa na získanie prehľadu hrozieb, ktoré môžu prinášať riziká. Nájdene hrozby môžu ohroziť stanovené ciele spoločnosti alebo byť prekážkou v ich dosiahnutí.
- Strengths – Silné stránky – ide hlavne o prednosti, ktoré spoločnosť má a je schopná ich využiť v prospech dosiahnutia stanovených cieľov.
- Weaknesses – Slabé stránky - sústredí sa hlavne na nedostatky a problémy, ktoré aktuálne v spoločnosti existujú a mohli by ohroziť stanovené ciele. [7]

#### 1.4.1.4 Porterova analýza piatich síl

Porterova analýza piatich síl sa sústreďí na analýzu konkurenčných síl okolia firmy a samotné postavenie spoločnosti v podnikateľskom prostredí. Táto analýza sa snaží identifikovať sily, ktoré v odvetvovom prostredí na firmu pôsobia, a pomocou nich identifikovať strategické príležitosti a hrozby konkurencie, ktoré môžu nastať. Snahou tejto analýzy by malo byť pre spoločnosť nastavenie takej pozície, v ktorej by sa firma mohla najlepšie brániť voči konkurenčným silám.



Obrázok 3: Porterova analýza piatich síl (Zdroj: [14])

Súčasťou Porterovej analýzy je päť hlavných vplyvov, ktoré ovplyvňujú konkurencieschopnosť podniku, a tými sú:

- Hrozba vstupu nových konkurentov - vplyv tohto rizika sa znižuje hlavne v spojení s nákladmi (ak sú veľmi vysoké pre vstup nového konkurenta na trh) alebo v prípade monopolov. V prípade nízkeho rizika tohto vplyvu môžu firmy ceny zdvíhať nezávisle na konkurencii, a docieľiť väčší zisk. [19]
- Hrozba substitútov – dôsledkom zvyšovania nahraditeľnosti produktu substitútom, sa znižuje atraktivnosť daného odvetvia. [20]
- Vyjednávací sila odberateľov – tento faktor je dôležitý z pohľadu identifikovania zákazníkov a ich potrieb, ktoré sa môžu meniť na základe

trendov. Trendy môžu viesť ku zmene požiadaviek a chovania, a preto je ich potrebné sledovať. [19]

- Vyjednávacía sila dodávateľov – tento vplyv je tým väčší, čím viac je spoločnosť na danom dodávateľovi závislá. V dôsledku toho je vhodné analyzovať závislosť na dodávateľoch, sledovať konkurenciu medzi dodávateľmi ako takými, a uvažovať nad prípadnou náhradou za existujúceho dodávateľa. [19]
- Rivalita medzi existujúcimi spoločnosťami - vplyv konkurencie patrí medzi najsilnejšie z piatich síl, ktoré ovplyvňujú postavenie spoločností v konkurenčnom odvetví. Rivalita medzi existujúcimi spoločnosťami je dôsledkom snahy o vylepšenie pozície každej firmy. [20]

#### **1.4.2 Projektová fáza**

Táto fáza obsahuje samotnú realizáciu projektu od jeho zahájenia, cez plánovanie až po samotnú implementáciu. Výsledkom tejto fázy je v najlepšom prípade dokončený projekt a zákazníkovi je odovzdané všetko, čo bolo sľúbené. Túto fázu môžeme rozdeliť na:

- Zahájenie projektu (start-up)
- Plánovanie
- Vlastná realizácia
- Ukončenie projektu (close-out)[7]

#### **1.4.3 Poprojektová fáza**

Dochádza k nej až po dokončení projektu. V rámci tejto fázy prebiehajú analýzy dokončeného projektu a konečné vyhodnotenie či bol tento projekt úspešný alebo nie. Sem patrí aj spracovanie návrhu do budúcich projektov na základe nadobudnutých vedomostí a skúseností z dokončeného projektu.[7]

### **1.5 Produkt Projektu**

Cieľom projektového snaženia je vytvorenie jedinečného produktu – predmetu, služby alebo ich kombinácie, ktorá naplní očakávania jednak zadávateľa projektu a na druhej



strane sa bude podieľať na dosiahnutí strategického cieľa, ktorý súvisí s jeho aktivitami.[3]

*„Produkt projektu je cieľ, výsledok alebo iný výstup projektu, ktorý má byť realizáciou projektu vytvorený.“ (3, s.24)*

Pre produkt projektu by mala platiť jedna alebo viacero z nasledujúcich možností:

- Je kvantifikovaný a môže predstavovať ucelený fyzický objekt alebo jeho časť
- Vytvára nejakú službu, ktorá zlepšuje výkonnosť podnikového procesu
- Vytvára vstup pre iné interné alebo externé procesy

V skutočnosti sa produktom môže stať čokoľvek, napríklad zavedenie existujúceho procesu alebo iných organizačných zmien v spoločnosti, realizácia nejakej novej stavby, štúdie, implementácia novej technológie atď. Preto je dôležité jasne pomenovať problém a jeho zadanie, pretože hrá významnú rolu od plánovania až po samotnú akceptáciu projektu. [3]

## **1.6 Časová analýza projektu**

Metódy sieťovej analýzy tvoria základ metód plánovania v súčasnom projektovom riadení, pričom využívajú techniku uzlovo orientovaných grafov, ktorá vychádza z matematickej teórie grafov. Najrozšírenejšou technikou, ktorá sa používa na odhadovanie celkovej dĺžky projektov a kritických miest (činností) sú uzlovo orientované sieťové grafy. K použitiu týchto metód je nutné si na začiatku stanoviť jednotlivé aktivity.[7]

Spracovanie časovej analýzy vyžaduje, aby každá činnosť mala priradený údaj o dĺžke jej trvania a prostredníctvom sieťového grafu je potom analyzovaná výsledná dĺžka trvania projektu a rezervy. Nutnosťou je na začiatku stanoviť jednotku, v ktorej budú uvádzané činnosti.[7]

Spoločným základom pre sieťovú analýzu je teória grafov a teória pravdepodobnosti. Tento typ analýzy sa používa na viacerých miestach, ktoré úzko súvisia plánovaním, koordináciou a kontrolou zložitých úloh a to vo viacerých oblastiach ako sú:

- Na úseku investičnej výstavby
- Výskum a vývoj

- Oblasť údržby základných fondov
- Zaisťovanie zložitých administratívnych úloh
- Strojárska výroba
- Plánovanie priebehu skúškového obdobia študenta [15]

### 1.6.1 Metóda kritickej cesty CPM

Critical path method (metóda kritickej cesty) je metóda založená zväčša na deterministickom modeli odhadovania. Pre výpočet sa využíva uzlovo orientovaná sieťová analýza. Hlavným cieľom tejto metódy je identifikovanie kritickej cesty.[15]

*Metóda CPM predovšetkým vypočítava tzv. Kritickú cestu. Kritická cesta je časovo najdlhšia cesta grafom a predstavuje najkratšiu dĺžku trvania projektu podľa aktuálneho plánu. (7, str.137)*

Pre kritickú cestu platia aj nasledujúce tvrdenia:

- RC jednotlivých činností sa rovnajú nule
- Je najdlhšia cesta v grafe
- Činnosti na tejto ceste sú dôležité a nemôžu sa oneskoriť [7]

Metóda CPM pracuje s hranovo definovanými sieťovými grafmi s konjuktívne deterministickou interpretáciou uzlov. Nutnosťou je rozloženie každej zložitej činnosti na niekoľko čiastočných činností, medzi ktorými existuje časová nadväznosť a podmienenosť. Umožňuje nám riešenie zložitých časových následností s cieľom dosiahnuť maximálne skrátenie celkového priebežného času, ktorý je potrebný na realizáciu celého projektu.[15]

### 1.6.2 Metóda PERT

Na rozdiel od CPM, PERT (Program Evaluation and Review Technique) pracuje s hranovo definovanými sieťovými grafmi, kde trvanie každej činnosti sa považuje za náhodnú veličinu s určitým rozdelením pravdepodobnosti.

Metódu PERT používame vždy tam, kde sú činnosti neopakovateľné a dobu trvania činnosti nevieme dopredu určiť. Časová analýza metódou PERT sa dá robiť dvomi spôsobmi, a to pomocou metódy Monte Carlo alebo prevodom na deterministický model. [15]

Čo sa týka deterministického modelu, tak na určenie trvania činnosti sa používajú tri časové odhady:

- Optimistický
- Najpravdepodobnejší
- Pesimistický[15]

Vzorce, ktoré sa pri metóde PERT využívajú sú zobrazené na obrázku nižšie. Jedným z nich je aj výpočet doby trvania činnosti vzhľadom na prevod na deterministický model.

**Střední doba trvání činnosti** (převod na deterministický model):  $y_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$

$$\text{Rozptyl: } \sigma^2 y_{ij} = \frac{(b_{ij} - a_{ij})^2}{36}$$

$$\text{Směrodatná odchylka: } \sigma y_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Jedna bunka v grafe je definovaná nasledujúcim obrázkom a pod ním sú popísané aj jednotlivé veličiny.

**Tabuľka 2: Uzol sieťového grafu PERT metódy** (Zdroj: vlastné spracovanie)

ZM	RC	KM
Činnosť		
ZP	RV	KP

#### **ZM – možný začiatok činnosti**

Určuje najskorší možný termín začatia danej činnosti. Hodnoty možného začiatku činností sa určujú vždy od počiatočného uzlu až po koncový.

#### **ZP- prípustný začiatok činnosti**

Určuje posledný možný termín, do ktorého je možné termín začatia činnosti odložiť bez toho, aby sa ohrozil termín ukončenia projektu. Výsledok stanovíme odčítaním času trvania činnosti od termínu najneskôr prípustného ukončenia činnosti.

#### **RC – rezerva celková**

Udáva čas, o ktorý sa dá činnosť najviac oneskoriť, bez toho aby sa celková doba trvania projektu zmenila. Hodnotu zistíme odčítaním KP a KM alebo ZP a ZM. Kritickú cestu tvoria činnosti, ktorých RC je rovná 0.

#### **RV – rezerva voľná**

Je to rezerva, o ktorú sa môže daná činnosť oneskoriť bez toho, aby sa nasledujúca činnosť oneskorila. Výpočet je možný odpočítaním ZP nasledujúcej činnosti a KM aktuálnej činnosti.

#### **KM – najskôr možný koniec**

Táto hodnota je daná súčtom najskôr možného termínu zahájenia činnosti a doby jeho trvania.

#### **KP – najneskôr prípustný koniec**

Najneskôr prípustný koniec určíme ako najnižšiu hodnotu zo všetkých ZP predchodcov aktuálnej činnosti.[15]

### **1.7 Riadenie rizík**

Výskyt rizika a neistoty je potrebné brať ako súčasť každého projektu. Výskum a vývoj nových produktov, zavádzanie nových technológií, vstupy na nové trhy, investície do veľkých projektov, reštrukturalizácie v rámci firmy, sú všetko činnosti, pri ktorých musíme počítať s určitým stupňom rizika.[16]

#### **1.7.1 Riziko**

Na zadefinovanie rizika neexistuje výhradne iba jeden pojem, ale je možné použiť viacero definícií. Na definovanie rizika sa používajú napríklad tieto definície:

Riziko je:

1. Pravdepodobnosť či možnosť vzniku straty.
2. Variabilita možných výsledkov alebo neistota ich dosiahnutia.
3. Odchýlenie skutočných a očakávaných výsledkov.
4. Pravdepodobnosť akéhokoľvek výsledku odlišného od výsledku očakávaného.[17]

Všeobecne sa však nemusí jednať len o riziká ekonomické, ale môžu byť tak isto spojené s politikou, teritóriom, bezpečnosťou alebo právom.

Tak isto je nutné rozlíšiť postoj manažéra alebo podnikateľa k riziku, a to na:

- Averzia – subjekt s averziou k riziku sa vyhýba rizikovým podnikateľským projektom a preferuje tie s istejšími výsledkami
- Sklon k riziku – subjekt vyhľadáva značne rizikové projekty, ktoré môžu byť ako ziskové, tak aj veľmi stratové
- Neutrálny postoj – osoba má rovnováhu medzi sklonom a averziou k riziku [17]

Z pohľadu finančnej teórie je riziko často zamieňané s kolísavosťou finančných veličín okolo očakávanej hodnoty v dôsledku zmien celej rady parametrov.[14]

Z podnikateľského hľadiska sa na riziká pozeráme z dvoch stránok:

- Pozitívne – nádej vyššieho zisku, nádej vyššieho úspechu
- Negatívne – nebezpečenstvo horších hospodárskych výsledkov [14]

### 1.7.2 Analýza rizika

Prvá vec, ktorá je potrebná pri znižovaní rizika je jeho analýza. Analýza je chápaná ako proces definovania hrozieb, pravdepodobnosti ich uskutočnenia a dopadu na aktíva, teda stanovenie rizík a ich závažností. [17]

Na začiatok je potrebné zadefinovať pojmy, ktoré sa úzko spájajú s analýzou rizika a budú v texte spomenuté.

**Aktívum** – je všetko čo má pre subjekt hodnotu, ktorá môže byť zmenšená pôsobením hrozby. Delíme ich na hmotné a nehmotné.[14]

**Hrozba** – v prípade hrozby sa hovorí o sile udalosti, aktivity alebo osoby, ktorá má nežiadúci vplyv na aktíva alebo môže spôsobiť škodu organizácii ako celku. Hrozby sa dajú členiť podľa rôznych kritérií. Škoda, ktorá môže vzniknúť pri pôsobení hrozby na dané aktívum, sa nazýva dopad hrozby.[17]

Hrozbu ako možnosť výskytu nepriaznivej udalosti môžeme vyjadriť pomocou pravdepodobnosti  $P$  a hodnotou nepriaznivého dopadu  $D$  na projekt. Ak tieto veličiny pre daný prípad poznáme potom je možné vypočítať hodnotu rizika nasledovne:

$$HR = P \times D$$

**Zraniteľnosť** – je nedostatok, slabina alebo stav analyzovaného aktíva, ktorý môže hrozba využiť v dôsledku nežiaduceho vplyvu.[14]

**Protiopatrenie** - sa dá definovať ako postup, proces, procedúra alebo technický prostriedok, ktorý bol navrhnutý na zmiernenie pôsobenia hrozby, zníženie zraniteľnosti alebo dopadu hrozby [14]

Do analýzy rizík zahŕňame:

1. Identifikácia aktív - vymedzenie posudzovaného subjektu a popis aktív, ktoré vlastní.
  2. Stanovenie hodnoty aktív - určenie hodnoty aktív a ich významu pre subjekt, ohodnotenie možného dopadu v prípade ich straty, zmeny či poškodenia na existenciu alebo chovanie subjektu.
  3. Identifikácia hrozieb a slabín – identifikácia druhov udalostí a akcií, ktoré môžu negatívne ovplyvniť hodnotu aktív alebo stanovenie slabých miest subjektu, ktoré môžu umožniť pôsobenie hrozieb.
  4. Stanovenie závažnosti hrozieb a miery zraniteľnosti – stanovenie pravdepodobnosti výskytu hrozby a miery zraniteľnosti subjektu voči hrozbe.
- [17]

Po vykonaní analýzy je potrebné na riziko nejako reagovať v závislosti na výsledkoch z analýzy. Na základe určenia hranice sa stanoví, či sa oplatí dané riziko podstúpiť alebo nie. Podľa stanovenej hranice sa riziká zoradia od kritických, dôležitých a menej dôležitých. V závislosti na klasifikácii rizík subjekt môže reagovať rôznymi spôsobmi.

1. Akceptácia rizika – jedná sa o prijatie a podstúpenie rizika a následné pokračovanie v projekte.
2. Retencia rizika – dá sa definovať ako podstúpenie rizika, pričom retencia môže byť dobrovoľná alebo nedobrovoľná a tak isto vedomá alebo nevedomá
3. Redukcia rizika – tu sa môže jednať o redukciu podnikateľskej aktivity alebo až redukciu konkrétnych dôsledkov podnikateľskej aktivity.
4. Presun rizika – prenesenie rizika na niekoho iného, zaraďuje sa medzi metódy s defenzívnym prístupom k riziku.[14]

Fáza, ktorá nasleduje po analýze sa nazýva sledovanie rizík, kde sa podľa potreby realizujú pripravené opatrenia ku zníženiu rizika alebo prebiehajú analýzy nových potencionálnych rizík. Spojenie analýzy rizík a sledovania rizík sa v rizikovom inžinierstve označuje ako riadenie rizík.[7]

Pri analýze rizík sa v rámci projektového riadenia používajú napríklad:

- Metóda RIPRAN
- Skórovacia metóda alebo tzv. bodovacia metóda s mapou rizík

Tieto metódy zaraďujeme medzi tzv. kvalitatívne metódy, pretože sú vyjadrené verbálnymi hodnotami. Oproti kvantitatívnym metódam sa líšia v tom, že kvantitatívne metódy ohodnocujú riziká pomocou konkrétnych číselných hodnôt (hodnota pravdepodobnosti a dopadu).[7]

### **1.7.3 Skórovacia metóda**

Vzhľadom na to, že v tejto diplomovej práci bude použitá práve táto metóda, tak bude v tejto podkapitole priblížený postup pri jej použití. Jedná sa o metódu, ktorej použitie je jednoduchšie ako pri metóde RIPRAN. Je vhodnejšia pre situácie v časovej tiesni a pre malé projekty.

V skórovacej metóde sa vyskytujú tri fázy a tými sú:

1. Identifikácia rizika
2. Ohodnotenie rizika
3. Návrhy opatrení k zníženiu rizika

Pre každú možnosť nebezpečenstva rizika sa v skórovacej metóde ohodnotí jednak pravdepodobnosť nebezpečnej udalosti tak aj jej dopad, prostredníctvom desaťbodovej stupnice. V tomto prípade je hodnota pravdepodobnosti nahradená určením možnosti výskytu predpokladanej udalosti alebo rizikového faktora. Každý člen v rámci projektového tímu by mal stanoviť svoj odhad. Výsledné skóre sa vypočíta ako aritmetický priemer pre pravdepodobnosť a pre dopad na projekt. Ocenenie rizika sa vypočíta ako súčin týchto aritmetických hodnôt v rozmedzí 1-100.[7]

Po ohodnotení rizík, je potrebné vyhodnotiť riziká na základe hodnôt získaných z predošlého postupu a prípadne na ne reagovať na základe navrhnutých opatrení. K vyhodnoteniu významnosti rizík sa používa mapa rizík s rozdelením do jednotlivých kvadrantov a to:

- Kvadrant bezvýznamných hodnôt rizík
- Kvadrant bežných hodnôt rizík
- Kvadrant významných hodnôt rizík

- Kvadrant kritických hodnôt rizík



Obrázok 4: Mapa Rizík - rozdelenie kvadrantov (Zdroj: [7])

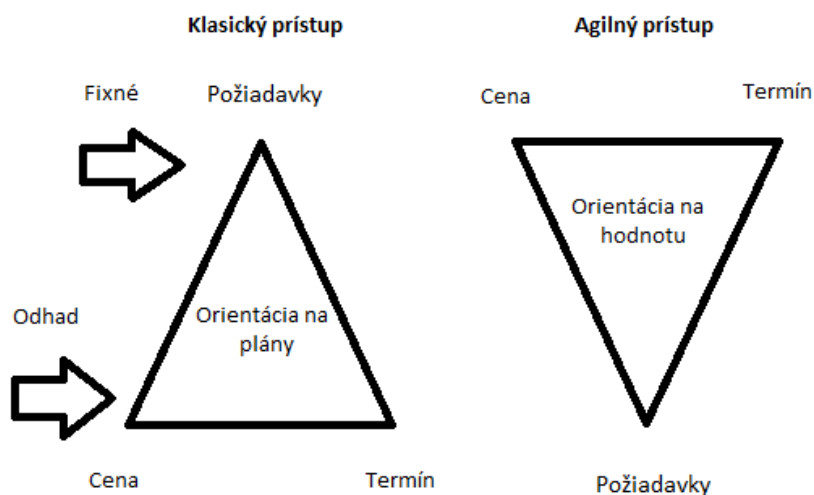
Na základe vyhodnotenia významnosti rizík z grafu a podľa toho, do ktorého kvadrantu riziká spadajú, sú navrhnuté vhodné protiopatrenia.

## 1.8 Tradičné metodiky vývoja projektu (softwaru)

Táto podkapitola sa venuje jednotlivým pojmom týkajúcim sa metodík vývoja projektov, týkajúcim sa software, nakoľko diplomová práca sa bude zaoberať IT firmou vyvíjajúcou software. Tradičné metodiky sa označujú ako tradičné kvôli tomu, aby sme ich boli schopný odlíšiť od dnešných moderných agilných metodík, ktorými sa budeme zaoberať v ďalšej podkapitole. Tradičné jednak znamená to, že tieto metodiky vznikli skôr (to však nemusí vždy platiť, pretože existujú aj novo vzniknuté tradičné metodiky) ale predovšetkým preto, že tieto metodiky uplatňujú tradičné prístupy k vývoju software. V tradičných metodikách je charakteristickou vlastnosťou to, že sa snažia obmedziť vágnosť a naopak čo najpresnejšie určiť jednotlivé termíny a požiadavky. Role sú väčšinou v týchto metodikách presne dané, čo znamená, že každý má svoju špecializáciu a neangažuje sa v tých činnostiach a fázach, ktoré mu nepatria.[11]

Na začiatku by bolo dobré definovať pojem software ako taký, aby sme vedeli o čom konkrétne píšeme.





**Obrázok 5: Klasický vs agilný prístup** (Zdroj: Vlastné spracovanie)

### 1.8.1 Software

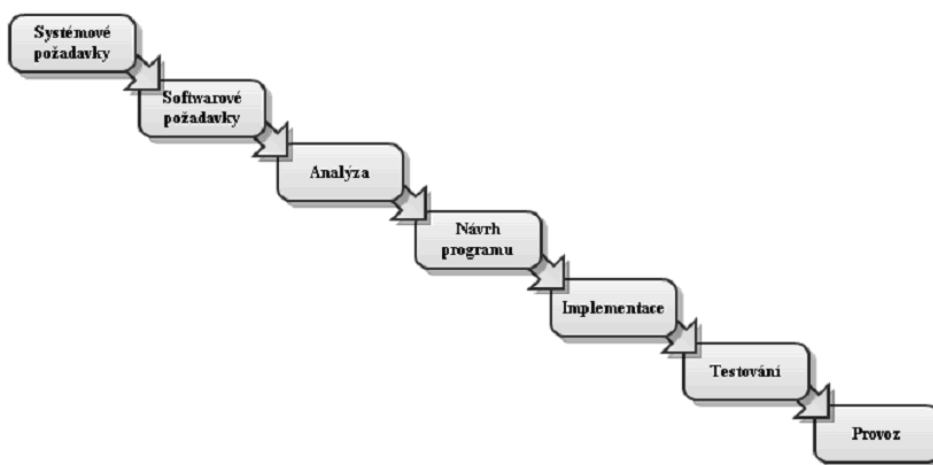
V predošlej podkapitole sme si definovali produkt projektu. Teraz si skúsime definovať produkt software ako takého.

Pokiaľ členovia projektového tímu vytvárajú software v zmysle výrobku pre daného zákazníka alebo užívateľa, nazývame takýto produkt softwarovým produktom. Softwarový produkt však nie sú len počítačové programy, ale softwarový produkt zahŕňa aj požiadavky, špecifikácie, popisy návrhu, zdrojové texty programu, testovacie dáta, príručky a manuály.[10]

### 1.8.2 Vodopádový model (Waterfall model)

Vodopádový model patrí medzi najstaršie metodiky vývoja software, ale pre väčšinu projektov je momentálne nedostačujúci. Vodopádový model je založený na jednoduchej úvahe. Softwarový proces, alebo proces vývoja software sa skladá z niekoľkých fáz, ktoré musia nasledovať jedna za druhou. A presne takto je možné proces riadiť a spravovať. Jednotlivé fázy vždy nasledujú po fázach predchádzajúcich, pričom sa nikdy nevracajú.

Základným problémom vodopádového modelu je užívateľ, ktorý nie je na začiatku schopný určiť základné požiadavky. Potom keď na konci dostane spustiteľnú verziu software, vznesie celý rad nových požiadaviek, prípadne opraví pôvodné. Takže namiesto toho, aby sa systém nasadil a používal, tak sa v podstate dostávame na začiatok, čo je v konečnom dôsledku problém, pretože je potrebné absolvovať všetky etapy od začiatku. Čiže ako najväčší problém tejto metodiky je to, že zákazník behom vývoja nezasahuje do vývoja a dostáva sa ku software až na konci, pri jeho odovzdávaní.[10]



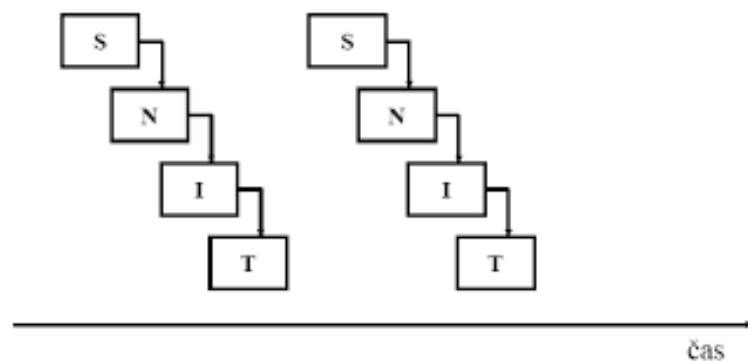
Obrázok 6: Vodopádová metóda (Zdroj: podľa [11])

Táto metodika sa dá charakterizovať podľa týchto bodov:

- Lineárny priebeh – každá fáza nasleduje po predchádzajúcej, nikdy sa nevracia.
- Jednoznačnosť – vždy je jasné v akej fáze sa projekt nachádza.
- Úplné zadanie – do ďalšej fázy sa vstupuje, keď je predchádzajúca fáza so svojimi výstupmi dokončená.[11]

### 1.8.3 Iteratívny model

Tento model bol vytvorený v dôsledku zlepšenia existujúceho vodopádového modelu, ktorý sme si popisovali v predošlej kapitole. Dôvodom bolo odstránenie hlavných problémov, ktoré sa v ňom objavili. Iteratívny model sa snaží odstrániť hlavný problém vodopádového modelu, ktorým je to, že užívateľ vidí spustiteľnú verziu softwaru príliš neskoro. Tento problém sa snaží riešiť iteratívne.



**Obrázok 7: Iteratívny model** (Zdroj: [10])

Každú iteráciu (kde jednotlivé písmená chápeme ako S = špecifikácia, N = návrh, I = Implementácia, T = testovanie) môžeme chápať ako jednu inštanciu vodopádového modelu. V podstate po každej iterácii sme zákazníkovi schopný dodať software, ktorý si je schopný odskúšať, a na základe toho sa môžu dodatočne v ďalšej iterácii spracovať jeho požiadavky. Jasnou výhodou tohto modelu je možnosť upravovať požiadavky užívateľom bez toho, aby to narušilo životný cyklus software. Naopak vývoj software je nutné rozdeliť do iterácií, čo môže viesť k návrhu horšej štruktúry systému.

Čiže základným princípom tohto modelu je opakovanie iterácií až do tej doby, než dosiahneme konečný cieľ, pričom prvá iterácia zvykne byť pomerne hrubá, druhá je detailnejšia a v poslednej iterácii už len ladíme najmenšie detaily podľa požiadaviek zákazníka.[11]

Iteratívny cyklus by sa dal popísať aj takto:

- Rozmyslite si čo potrebujete (špecifikácia požiadaviek)
- Urobte potrebné merania (Analýza)
- Navrhnete postup danej práce (Návrh)
- Urobte zadanú úlohu (Implementácia)
- Skontrolujte, či je splnený cieľ (Testovanie)
- Pokiaľ nie, vraciate sa k bodu 1.[11]

#### **1.8.4 Inkrementálny model**

Inkrementálny model je upravený iteratívny model s tým, že na základe špecifikácie sa určia ucelené časti systému, ktoré sa odovzdávajú užívateľovi. Oproti iteratívne

modelu je výhoda tá, že štruktúra môže byť na začiatku navrhnutá lepšie ako pri iteratívnom modeli.[10]

### **1.8.5 Špirálový model**

Spolu s inkrementálnym modelom sa zaraďujú medzi iteratívne metódy, pre ktoré platia tieto podmienky:

- Spustiteľný kód po každej iterácii – zákazník po každej iterácii vidí čiastočný výsledok, ktorý si môže vyskúšať a prípadne vyjadriť svoje pripomienky.
- Testovanie ako súčasť každej iterácie – chyby sú odhalené v rámci každej iterácie a potom sú opravené v nasledujúcej iterácii.
- Zmeny sú súčasťou procesu – v každej iterácii zákazník jednak testuje aplikáciu za účelom odhaľovania chýb, ale taktiež aj to či systém spĺňa jeho požiadavky.[11]

Tento model nám zavádza do procesu vývoja software nový pojem, a to je prototypovanie a kladie značný dôraz na analýzu rizík. Tu však užívateľ nedostáva na vyskúšanie určitú verziu softwaru s ohraničenou funkcionalitou, ale len prototyp.

Prototyp sa na rozdiel od verzie s obmedzenou funkcionalitou líši v tom, že prototyp sa po použití zahodí a software sa vytvára znovu. To prispieva k tvorbe čistejšieho a jednoduchšieho systému.[10]

Pri použití tohto modelu sú už od začiatku dostupné spustiteľné verzie, takže chyby sú odhalené hneď na začiatku, čo znižuje cenu za ich odstránenie. Avšak špirálový model má hlavnú nevýhodu v tom, že je náročný na riadenie a dá sa len ťažko získať celkový prehľad o projekte.[10]

## **1.9 Agilné metodiky**

V predošlej kapitole sme si opísali tradičné metodiky. Tak ako sme si ich opísali, jedná sa o metodiky, ktoré sú tak nazvané nie kvôli tomu, že by boli o toľko novšie ale hlavne kvôli prístupu k tvorbe software. Samozrejme, že existujú projekty, kde tradičné prístupy nie sú úplne to pravé.[11]

### 1.9.1 Čo znamená agilne?

Slovo agilný v doslovnom preklade znamená dynamický, interaktívny, zábavný, hravý alebo rýchlo reagujúci. Jedná sa o uprednostňovanie iných hodnôt: reálneho výsledku pred striktnými procesmi alebo zmenou pred dopredu naplánovanou vecou. Ale presný návod ako sa stať agilným neexistuje, a tak isto ani žiadny agilný certifikát nepomôže k tomu správne chápaniu tejto metodiky. K tomu aby agilný vývoj fungoval, je potrebné agilne myslieť a agilne sa správať, čo tak isto súvisí aj s firemnou kultúrou a náladou.[12]

Agile je hlavne o komunikácii, spolupracovaní a pripravenosti na zmenu. Hlavnou myšlienkou je robenie toho, čo má v daný okamih zmysel. Agilný princíp má svoje pravidlá a nejedná sa o žiadny chaos. Necháva väčšiu voľnosť samotným tímom za účelom zdvihnutia produktivity, kvality produktu v čo najmenšom čase.[12]

Zmena tradičných metodík na agilné nemusí byť vždy to pravé, ale rozhodne nám poznanie agilných princípov rozširuje možnosti spôsobu vývoja software. Agilný prístup sa snaží eliminovať byrokráciu, zbytočné dokumentovanie každej aktivity, či udalosti a hlavne zjednodušuje proces zmeny. Bola by obrovská chyba túto zmenu chápať ako zmenu násilnú, či za každú cenu. Agilný prístup neznamena neporiadok, ale práve naopak poriadok, ako je to u tradičných metodík. Znamená iný prístup k poriadku, iný prístup k rozumne chápaným procesom a iný prístup k efektívnosti.[12]

### 1.9.2 Agilný manifest

Základom agilného prístupu je agilný manifest (Agile Manifesto). V podstate sa nejedná o žiaden oficiálny dokument ani zákon alebo predpis ale o prehlásenie, ktoré spísala skupina ľudí, ktorá sa venovala vývoju software a verili, že táto činnosť sa dá zlepšiť. Uvedieme formu agilného manifestu tak, ako bola pôvodne formovaná svojimi zakladateľmi.[12]

*„Objavujeme lepšie spôsoby vývoja software tým, že ich tvoríme a pomáhame pri jeho tvorbe ostatným. Pri tejto práci sme dospeli k týmto hodnotám:*

- ***Jednotlivci a interakcie pred procesmi a nástrojmi,***
- ***Fungujúci software pred vyčerpávajúcou dokumentáciou,***

- *Spolupráca so zákazníkom pred vyjednávaním o zmluve,*
- *Reagovanie na zmeny pred dodržiavaním plánu“(11, s.31)*

K lepšiemu pochopeniu jednotlivých tvrdení si ich skúsime priblížiť.

1. Jednotlivci a interakcie pred procesmi a nástrojmi – procesy a nástroje sú dôležité avšak pre agilné metodiky sú dôležitejší jednotlivci, ich potreby a ich vklad do projektu. Podobne aj interakcie. Nástroje a procesy slúžia k tomu aby uspokojili jednotlivca a aby sa uskutočnili jednotlivé plány. Tradičné metodiky väčšinou naliehajú na nástroje a procesy, ktoré sú dané, aj keď ľudia nie sú spokojní. To je obrovský rozdiel v tom, ako k tomu pristupuje agile.[11]
2. Fungujúci software pred vyčerpávajúcou dokumentáciou – hovorí o tom, že fungujúci software je v každom prípade primárnym cieľom a nie zdĺhavá dokumentácia. V žiadnom prípade agile nehovorí o tom, aby sa nedokumentovalo, ale aby sa dokumentovalo prirodzene a na základe toho, akú činnosť vývojár v danom momente robí, pretože pre zákazníka je dôležitý výsledný software.[12]
3. Spolupráca so zákazníkom pred vyjednávaním o zmluve – toto tvrdenie nám hovorí o preferovaní komunikácie so zákazníkom na úrovni normálnej ľudskej komunikácie pred komunikáciou prostredníctvom zmlúv. Štandardná komunikácia pomôže zákazníkovi lepšie špecifikovať čo konkrétne od výsledného produktu potrebuje a vývojový tím, ktorý software vytvára lepšie chápe čo zákazník potrebuje. V konečnom dôsledku je to prospešnejšie aj pre zákazníka a aj pre tím, ktorý projekt vyvíja.[11]
4. Reagovanie na zmeny pred dodržiavaním plánu – tak ako svet, aj technológie sa hýbu dopredu a tak môže nastať situácia, že konkurencia už prišla na trh s niečím novým, prípadne sa objavili nové lepšie riešenia daného problému. Je potrebné, aby sa na tieto zmeny reagovalo už počas vývoja, a aby to pre zákazníka vytvorilo žiadanú hodnotu, ktorú na začiatku očakával. Pre obidve strany je to lepšie riešenie ako dodržiavanie plánov za každú cenu.[12]

### 1.9.3 Dôvody prechodu na agile

V predošlej podkapitole sme si definovali hlavné princípy agile a hlavne jeho rozdiely v prístupe oproti tradičným metodikám. Teraz by bolo dobré spomenúť hlavné dôvody prechodu na agile, ktoré si postupne vysvetlíme.

- **Flexibilita** – u tradičných metodík bol väčšinou release, jedenkrát za dlhšiu dobu (8-10 mesiacov) podľa toho, ako sa to dopredu dohodlo, ale momentálne chce vidieť každý výsledky v čo najkratšej dobe, čo je u tradičných metodík problém.
- **Efektivita** – väčšinou platí, že dvaja alebo skupina ľudí dosiahne lepšie výsledky a v kratšom čase, ako keď na danej úlohe pracuje jednotlivec. Preto sa napríklad používa aj technika párového programovania, ktorá prináša lepšie výsledky. Preto sa agile koncentruje na prácu v tímoch, ktorý má jasný spoločný cieľ a rieši ho spolu. Ďalšou dôležitou vecou je sústrediť sa pri vývoji hlavne na veci, ktoré zákazník požaduje, využije, prinášajú mu najväčšiu hodnotu a tým ostatným neprikladať až takú dôležitosť.
- **Predvídateľnosť** – agilné metódy používajú iný spôsob odhadovania, a to v relatívnych jednotkách a do odhadovania sa zapája celý tím. Presnosť výsledkov odhadovania sa časom výrazne zlepši. Dôležitou časťou je aj rozdelenie projektu na menšie časti. Na kratšom úseku je úsilie tímu stabilnejšie a bez väčších výkyvov.
- **Kvalita** – V tomto smere je dôležitý zákazník, s ktorým sa komunikujú priebežné výsledky a vyvarujeme sa situácie, keď pri odovzdaní hotového produktu zákazník trvá na jeho prepracovaní z dôvodu nesplnenia jeho očakávaní. Na druhej strane je tím zodpovedný za kvalitu produktu a tým vzrastá dlhodobá udržiavateľnosť kódu.
- **Zábava** – aj samotný vývojári budú viac chápať čo, a prečo tak píšú. Budú lepšie chápať zákazníkove požiadavky, a tak isto zmysel produktu. A taktiež spolupracovať s ostatnými je zábavnejšie.[12]

#### 1.9.4 Typy agilných metodík

Aj keď sa v tejto diplomovej práci budeme prevažne venovať jednej metodike, spomenieme si ich v tejto podkapitole viac, aby sme vedeli, že existujú a vždy pri zavádzaní agile existuje viac metodík, ktoré sú na výber.

Medzi známe agilné metodiky patria:

- Extrémne programovanie
- Vývoj riadený vlastnosťami (Feature Driven Development)

- Vývoj riadený testami (Test Driven Development)
- Lean Development
- Kanban
- Scrum

Ako som už spomínal vyššie, v tejto diplomovej práci sa budeme prevažne venovať jednej metodike, a tou bude SCRUM.

### 1.9.5 Lean

Je to v podstate proces, ktorý bol prebratý z továrenskej výroby. Podstatou je, robiť veci len keď sú potrebné. Často sú používané názvy ako „systém ťahu“ alebo „just in time“. Lean je skôr o prístupe ako o striktných procesoch, tak ako to platí všeobecne v agilných metódach. Veľa firiem používa túto metódu napriek tomu, že ich zamestnanci často nepoznajú hlavné princípy. Kľúčom je naopak správne pochopiť filozofiu tejto metódy, a tou je jednoducho povedané zjednodušenie práce o to, čo by nemuselo prinášať hodnotu.

Medzi najznámejšími firmami využívajúcimi lean metódu je napríklad Toyota. Namiesto vyrábania a plnenia skladov kdekoľvek a kedykoľvek, došlo k výrobe dielov len keď boli naozaj potrebné. To znamená, že proces, ktorý dovtedy fungoval vo väčšine firiem bol zrazu úplne obrátený.[11]

A pri vývoji softwaru sa to dá uplatniť tak isto ako pri továrenskej výrobe. Cieľom je obmedziť rozpracovanú prácu a sústrediť sa na to, aby boli jednotlivé požiadavky dokončené včas. Predpokladom je nezačínať s analýzou pokiaľ neexistuje prioritná požiadavka od zákazníka.

Pre Lean Development platia nasledovné princípy:

- Odstrániť všetko čo neprináša hodnotu
- Zlepšovať sa a učiť už v priebehu práce
- Rozhodovať sa čo najneskôr
- Dodávať prácu tak rýchlo ako sa dá
- Dať tímu dôveru a zodpovednosť
- Zamerať sa na celkový výsledok [11]

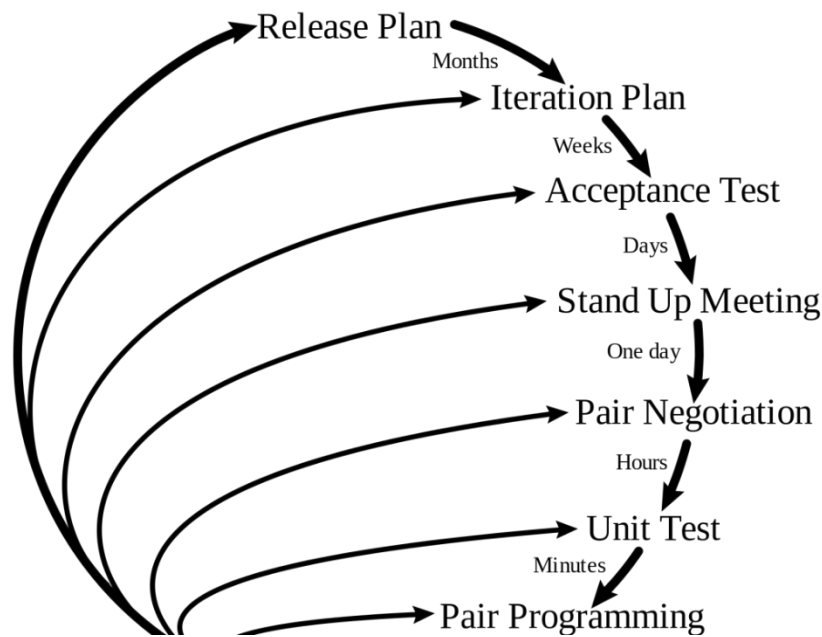


### 1.9.6 Extrémne programovanie

Extreme programming sa hlavne orientuje na veci, ktoré sa osvedčili. To v skratke znamená, že keď niečo fungovalo dobre, prečo to potom nerobiť stále tak isto. Táto metóda úzko súvisí so spätnou väzbou od zákazníka. Zmeny v projekte sú možné aj v neskorších fázach vývoja na základe komunikácie zákazníka s vývojovým tímom. Vývoj je postavený hlavne na tímovej práci. Samotné tímy sú zložené nielen z vývojárov a testerov, ale aj z manažérov a zástupcov zákazníka. Tímy sú väčšinou samoorganizované, postavené na hodnotách komunikácie, jednoduchosti, spätnej väzby, rešpektu a odvahy. Všeobecne sa preferuje jednoduchý a čistý návrh pred zložitým kde sa testovanie začína od prvého dňa. Produkt je dodávaný po menších častiach, čo najskôr ako je to možné. Na grafe nižšie sú zobrazené jednotlivé fázy extrémneho programovania tak ako na seba nadväzujú.[11]

Základné hodnoty extrémneho programovania:

- Jednoduchosť – práca, na veciach ktoré sú momentálne potrebné. Nie je potrebné robiť nič navyš. Tím pracuje na niečom, čo ho baví, prináša to hodnotu a je tento produkt schopný udržiavať.
- Komunikácia - každodenná komunikácia od návrhu cez kód až po testovanie.
- Spätná väzba – v malých iteráciách je dodávaný funkčný software a tým sa ukazuje implementovaná funkcionalita. Na základe spätnej väzby sú implementované prípadné zmeny. Podstatné je že proces sa prispôbuje produktu a nie naopak.
- Rešpekt – rešpektovanie sa v rámci tímu navzájom a tak isto zákazníka patrí medzi základné stavebné kamene.
- Odvaha – táto hodnota bezpodmienečne patrí k tejto metóde. Vecné pomenovávanie problémov a ich okamžité riešenie, nezávisle na tom, ako sú náročné a nepríjemné. V prípade zlého odhadu prijať chybu a nebáť sa zmeny.[12]



Obrázok 8: Extrémne programovanie (Zdroj: [12])

Extreme programming zameriava prvé týždne na testovanie spike solutions. Spike solutions sú v podstate riešenia častí architektúry, technológie alebo designu, ktoré by mohli byť problematické. Testovanie týchto prvotných riešení je dôležité z pohľadu obmedzenia a spresnenia odhadov a technických problémov počas implementácie. Tieto testy sa počas implementácie už nepoužívajú a zahadzujú sa.[12]

### 1.9.7 Kanban

Pôvodom pochádza z Japonska a je využívaný najmä na miestach, kde sa Scrum a Extreme programming úplne nehodí. Kanban je využívaný hlavne v oblastiach operations a maintenance tímov, ale využívaný je aj v IT, ale pri porovnaní s ostatnými procesmi narážame na problém, že Kanban ako proces nám nič nenariaďuje a viacmenej necháva rozhodnutia na nás. Stačí dodržať tieto princípy:

- Obmedziť rozpracovanú prácu – work in progress
- Minimalizovať čas prechodu – lead time
- Vizualizovať progres [12]

Podstatou je rozdelenie procesu na jednotlivé fázy (Backlog, In Progress a Done) a obmedzenie počtu procesov v jednotlivom stĺpci. Keď dôjde k tomu, že do stĺpca nie je možné pridať ďalšiu úlohu, je najskôr potrebné dokončiť niektorú z rozpracovaných úloh. Základom je správne určenie limitov jednotlivých front v stĺpcoch. Takmer vždy

sa Kanban kombinuje s niektorými ďalšími prvkami, ktoré sú súčasťou Scrumu a tak isto Scrum využíva najmä obmedzovanie úloh v stĺpcoch, ktoré je typické pre Kanban.[12]

### 1.9.8 Scrum

Tento druh agilnej metodiky sa dá využívať aj na iných projektoch ako software. Kapitál každej firmy je tvorený rôznymi hodnotami a každý manažér považuje za najväčšiu hodnotu niečo iné. Medzi takéto hodnoty zaraďujeme know-how, patenty, rôzne formy priemyselného vlastníctva, ktoré chránia samotné produkty alebo služby, unikátne technológie v rámci daného štátu alebo aj financie. Potom je tu ešte jedna možnosť, na ktorú veľa manažérov a riaditeľov zabúda, a to sú ľudské zdroje. Bez kvalitných ľudí sa veľa úspechov nedostaví aj napriek veľkému úsiliu manažéra. Koniec koncov aj ostatné hodnoty (patenty, know-how alebo unikátne technológie), ktorým väčšina manažérov prikladá väčší význam ako samotným ľudským zdrojom, nie je možné dosiahnuť bez kvalitných ľudí. Práve preto sú v scrume ľudské zdroje tak dôležité a stavajú sa v samotných hodnotách medzi tie najdôležitejšie, čo firma má a mala by na nich od začiatku stavať. Preto je aj zostavenie projektového tímu dôležité a nepatrí medzi najľahšie úlohy pre manažéra, ale určite je základom úspešného projektu.[11]

Scrum Development Process je agilná metodika vývoja software, ktorej cieľom je predovšetkým zvýšenie efektivity pri vývoji software. Scrum inklinuje k objektovo orientovanému videniu sveta, pričom organizuje a riadi procesy novým spôsobom. Názov vznikol z anglického slova Scrum, ktorého význam pochádza z ragby a označuje skrumáž, mlyn niekoľkých hráčov na jednom mieste za účelom dotlačenia lopty do cieľovej pozície. Tento význam sa dá použiť pri vývoji software, kde sa dotlačením lopty do požadovanej pozície rozumie dokončenie aplikácie do podoby, akú zákazník požaduje.[18]

Scrum sa dá považovať za komplexný framework, ktorý je využívaný na riešenie komplexných problémov za účelom dodania produktu s čo najvyššou kvalitou.

Scrum je:

- Pomerne povrchný čo sa týka pravidiel – má ich iba zopár
- Ľahký na pochopenie

- Pomerne ťažký na riadenie [18]

Scrum nie je správne chápať len ako proces, techniku alebo metódu, ale hlavne ako efektívne spojenie a spoluprácu manažmentu a pracovných techník za účelom neustáleho zlepšovania produktu, ale tak isto aj tímu a pracovného prostredia. [18]

Podstata Scrumu pozostáva zo scrum tímov, v ktorých sú jasne dané role, udalostí, pravidiel a jednotlivých artefaktov. Pravidlá určujú, ako jednotlivé časti scrumu medzi sebou spolupracujú a komunikujú. Ako už bolo spomenuté, pravidiel v scrume nie je až tak veľa ako pri tradičných metódach a sú jasne definované.[18]

Hlavnou myšlienkou scrumu je znalosť pochádzajúca z predchádzajúcich skúseností a rozhodnutí založených na získaných vedomostiach z minulosti. Scrum je založený na iteratívnom a inkrementálnom prístupe v snahe čo najviac optimalizovať predvídanie a kontrolovať riziká. Každá implementácia empirického procesu pozostáva z troch hlavných pilierov a tými sú:

1. Transparentnosť – dôležité aspekty celého procesu musia byť viditeľné pre všetkých, ktorý zodpovedajú za konečný výsledok. Práve transparentnosť hovorí o tom, aby boli tieto aspekty definované spoločným štandardom chápané všetkými pozorovateľmi rovnako.
2. Kontrola – kontrola by mala prebiehať tak často aby bolo možné identifikovať odchýlky od konečného cieľa. Kontrola by naopak nemala byť až tak častá, aby nezabraňovala celému procesu. Najefektívnejšie je, ak je kontrola vykonávaná skúsenými inšpektormi v každom kroku procesu.
3. Prispôsobenie – ak sa v priebehu kontroly zistí, že niektoré aspekty sa odchyľujú z prijateľných hraníc a ovplyvňuje to výsledný produkt, je nutné reagovať a prispôbiť ďalší vývoj požadovanému cieľu. Adaptáciu je vždy nutné urobiť čo najrýchlejšie, aby sa predišlo väčším odchýlkam.[18]

V scrume sa používajú štyri hlavné udalosti na kontrolu a adaptáciu a tými sú:

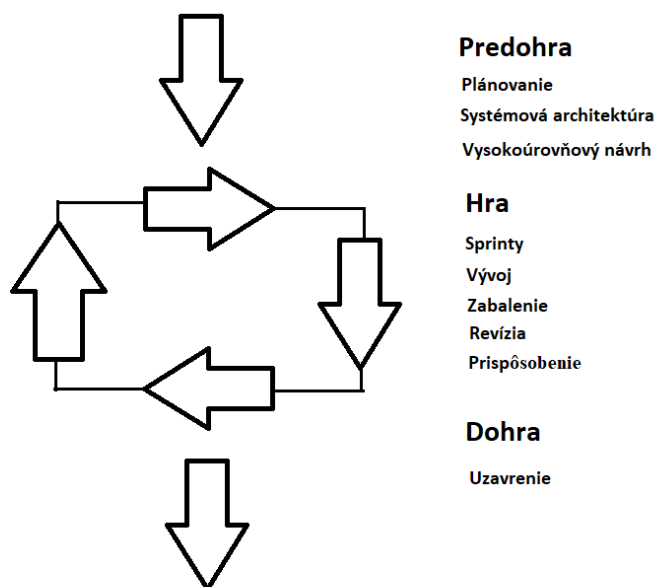
- Plánovací meeting,
- Daily scrum,
- Sprint review,
- Retrospektíva.[18]

Tieto udalosti budú popísané a bližšie vysvetlené v ďalších kapitolách tejto diplomovej práce.

#### **1.9.8.1 Postupnosť krokov a fáz v scrume**

Metodika Scrum definuje štyri vývojové kroky. Prvé dva patria do fázy predohry (Pregame), tretí do fázy hry (Game) a štvrtý do fázy dohry (Postgame):

1. Plánovanie – v tejto fáze sa definuje rozsah aktuálnej verzie, harmonogram, nutné zdroje atď. V rámci plánovania sa vyberajú úlohy, ktoré je nutné vyriešiť. V tejto časti sa robí aj analýza rizík a volia sa vhodné vývojové nástroje.
2. Architektúra a design – vytvorenie alebo modifikovanie architektúry v závislosti na nových požiadavkách a poznatkoch.
3. Vývoj – vývojové práce prebiehajú v iteratívnych cykloch pričom dĺžka jednej iterácie je ovplyvnená viacerými faktormi ako sú komplexnosť produktu, množstvo rizík, veľkosť tímu atď. Na konci každej iterácie sa zhromažďujú všetci členovia tímu, vedenie a zákazník kde sa prezentuje nový (funkčný) prototyp. Na tomto stretnutí sa definujú položky, na ktorých sa bude pracovať v ďalšej iterácii.
4. Uzavretie – táto fáza nastáva v prípade, ak vedenie alebo zákazník usúdia, že parametre aktuálnej verzie sú postačujúce. V tomto prípade sa nepokračuje s ďalšou iteráciou ale začína sa fáza uzavretia s finálnou prípravou produktu na uvoľnenie. V tejto fáze prebieha aj komplexná integrácia, testovanie a vytváranie dokumentácie.[18]



**Obrázok 9: metodika SCRUM Development Process** (Zdroj: vlastné spracovanie podľa [12])

#### 1.9.8.2 Role v Scrume

Okrem samotných členov podieľajúcich sa na projektoch je takisto dôležitá ich rola, ktorú budú vykonávať. Tento pojem môžeme vysvetliť viacerými spôsobmi, ale ako najpresnejšia bude rola, ako súbor práv, povinností, očakávaného jednania a chovania, ktoré úzko súvisí s tým, že dotýčný jedinec aktuálne zastáva svoju rolu. Oproti bežnému životu, kde sú role prevažne voľnejšie, metodiky role definujú. To znamená, že v rámci projektov sú role jednoznačne definované.[11]

V tradičných metódach vývoja software sú väčšinou jednotlivé role jasne dané a sú prísnejšie posudzované, čo sa prejavuje hlavne na deadlinoch a zodpovednostiach vyplývajúcich z jednotlivých rolí. V agilných metódach, kde SCRUM určite patrí, je rolí oveľa menej a sú oveľa univerzálnejšie definované, ako je tomu pri metódach tradičných. Napriek tomu, každý člen projektu má rolu vždy jednoznačne danú a je možné ju konkrétne určiť. Je dôležité rozlišovať medzi dvoma hlavnými rolami v Scrume:

- manažérske,
- výkonné.[13]

## **Manažérske role**

Dôležitou vecou pre ľudí na manažérskych postoch je autorita, ktorá sa nedá vynútiť ale dá sa postupne vybudovať. Naopak niektoré vlastnosti sa dajú identifikovať pomerne dobre. Hlavnými vlastnosťami človeka na manažérskom poste sú:

- odborná zdatnosť,
- morálny kredit,
- rešpekt k podriadeným,
- spravodlivý prístup,
- pevné názory.[12]

Manažérska rola v Scrum predstavuje múdre a uvedomelé vedenie ľudí schopných samostatnej práce a zároveň tímovej kooperácie. Napriek tomu manažér má stále aj bežné kompetencie, no musí akceptovať, že metodika je založená na demokratickom rozhodovaní ľudí.[12]

## **Výkonné role**

Druhým typom, ktoré sa v Scrum vyskytujú, sú výkonné role. Jedná sa o role, kde je hlavnou náplňou vykonávanie konkrétnych činností. V Scrum je jediná výkonná rola a tá sa nazýva člen tímu. Od člena tímu sa očakáva:

- odbornosť – je najdôležitejšia vlastnosť, očakáva sa dostatočná technická vyspelosť a skúsenosti,
- všestrannosť – schopnosť reagovať na rozličné problémy, nielen na veci, ktoré sú doménou daného človeka,
- samostatnosť – schopnosť vytvoriť riešenie a naimplementovať ho,
- flexibilita – schopnosť akceptovať meniace sa požiadavky a učiť sa nové technológie,
- schopnosť pracovať v tíme – Scrum je založený na úzkej tímovej spolupráci,
- ochota myslieť – schopnosť ponúknuť rozumné riešenie. [12]

## Pigs and Chickens

V Scrum sa ľudia, ktorý sa na projekte podieľajú, rozdeľujú do dvoch hlavných skupín a tými sú Pigs (prasatá) a Chickens (sliepky) . Pomenovania týchto rolí sú prevzaté z komiksového príbehu, kde sa prasa a kura dohadujú na pomenovaní názvu reštaurácie. Tieto role sú analogicky prevzaté pre a aplikované pri rozdelení ľudí, ktorý sa točia okolo projektu. Rozdelenie jednotlivých rolí s ich popismi je uvedené v tabuľke nižšie.[12]

**Tabuľka 3: Pigs and Chickens** (Zdroj: vlastné spracovanie podľa [12])

<b>Pigs</b>	<b>Chickens</b>
<b>Popis:</b> <ul style="list-style-type: none"><li>• pracovníci priamo zahrnutí v projekte</li><li>• podieľajú sa na projektových úlohách</li><li>• sú súčasťou projektového tímu</li><li>• sú priamo zodpovední za výsledok projektu</li></ul>	<b>Popis:</b> <ul style="list-style-type: none"><li>• pracovníci, ktorých sa projekt týka, ale nie sú doňho priamo zapojení</li><li>• nepriamo ovplyvňujú projekt a jeho výsledok (softwarový produkt)</li><li>• nie sú priamo zodpovední za výsledok</li></ul>
<b>Role:</b> <ul style="list-style-type: none"><li>• Product owner</li><li>• Scrum Master</li><li>• Scrum Team Member</li></ul>	<b>Role:</b> <ul style="list-style-type: none"><li>• Koncový užívateľ</li><li>• Člen vedenia organizácie zákazníka</li><li>• Konzultant</li></ul>

V tabuľke je popísané rozdelenie rolí v Scrum s ich popismi činností v projekte. Scrum sa snaží obmedziť množstvo vysoko špecializovaných rolí a snaží sa definovať pracovné činnosti tak, aby boli jednotliví pracovníci čo najviac univerzálni a aby boli vzájomne zastupiteľní.[12]

V nasledujúcej podkapitole budú postupne vysvetlené konkrétne jednotlivé role v scrume a tak isto aj pojmy, ktoré zo scrumom úzko súvisia.



### 1.9.8.3 Scrum tím

Scrum tím je nielen self-organized, ale tak isto multifunkčný a vzájomne zastupiteľný. V rámci tímu je dôležité rozdelenie práce a vždy pracovať najefektívnejšie ako tím, a nie ako jednotlivec. Preto, ak je niekto odborník napríklad na databázy, nezaobrá sa len databázami ale aj vecami, ktoré s databázami nesúvisia. V rámci scrum tímu sa nie je ťažiskom optimálne vyťaženie jednotlivca, ale optimálne nastavenie celého systému a jeho flexibilita pri reakcii na zmeny. [12]

Scrum tím sa skladá z vlastníka produktu (Product owner), vývojového tímu (Development team), Scrum mastra. Samo-organizujúce tímy na rozdiel od normálnych tímov v tradičných metódach si delenie a plánovanie práce robia sami a nie sú riadené ľuďmi mimo tímu. V Scrum je model tímu navrhnutý tak, aby optimalizoval flexibilitu, kreativitu a produktivitu. [18]

### 1.9.8.4 Vlastník produktu (Product owner)

Táto rola je veľmi dôležitá, pretože v podstate reprezentuje zákazníka. Názov tejto role je pomerne výstižný, pretože Product owner definuje, ako bude produkt vyzeráť a tak isto definuje aj jeho funkcionality. Môže ale nutne nemusí, byť zamestnancom zákazníka. Jeho dôležitou úlohou je, že definuje víziu celého projektu. Spolu so zákazníkom definuje, kam má projekt smerovať, čo je jeho cieľom a čomu sa má naopak celý projekt vyhnúť. Vízia je v rámci projektu veľmi dôležitá, bez nej bude projekt len ťažko úspešný. Preto je úloha vlastníka produktu pomerne zložitá, lebo potrebuje odlišiť výsledný produkt od ostatných. [12]

Ďalšími dôležitými úlohami vlastníka produktu sú:

- Stanoviť víziu projektu a jeho cieľov.
- Vytvorenie zoznamu úloh na základe požiadaviek od zákazníka.
- Určenie priorít týchto úloh.
- Udržiavať hodnotu produktu počas vývoja.
- Sprostredkuje komunikáciu medzi zákazníkom a vývojármi.
- Udržiava zoznam úloh viditeľný, transparentný, a definuje vývojovému tímu na čom bude pracovať.
- Zaisťuje, že vývojový tím rozumie definovaným úlohám.

- Má právo zrušiť Sprint – musí byť doložený faktami, že na projekte sa neoplatí pokračovať alebo to nemá zmysel. [18]

#### **1.9.8.5 Scrum Master**

Druhou významnou a povinnou rolou v rámci Scrumu je rola Scrum Mastra. Zjednodušene povedané sa jedná o manažérsku rolu, ale nie v takom zmysle, v akom to väčšinou chápeme. Jeho úlohou je vytvoriť tímu čo najlepšie podmienky na prácu, nakoľko tímy v scrume sú schopné pracovať bez klasického šéfovania. Na rozdiel od vlastníka produktu, ktorý úlohy definuje tvrdým spôsobom, Scrum Master by mal disponovať mäkkými zručnosťami. On je väčšinou tým členom, ktorý podporuje, motivuje a chráni. [12]

Základné úlohy scrum mastra by sa dali definovať takto:

- Pomáha tímu dosiahnuť ciele.
- Odstraňuje a rieši problémy.
- Motivuje tím k lepším výsledkom.
- Chráni tím pred vonkajšími vplyvmi, ktoré by ho odvádzali od sústredenia na definovanom ciele. [13]

Takže scrum master tím neriadi, ten sa riadi sám. Aj keď tím neriadi priamo, tak svojou autoritou smeruje tím tam, kde ho chce mať, a to je v rámci koncepcie Scrumu presne úloha, ktorú Scrum master zastáva. [11]

#### **1.9.8.6 Scrum Team Member**

Poslednou z troch hlavných rolí scrum tímu je Scrum Team Member, teda „radový“ člen tímu. Jedná sa o pracovníka, ktorý má veľkú voľnosť v plnení svojich úloh. Ani Product owner a ani Scrum Master nehovoria radovému členovi ako má svoje úlohy plniť (v rámci technológií a technických parametrov). Preto je považovaný za veľmi dôležitú osobu v tíme. V Scrume sa nerozlišujú pozície ako architekt, analytik, tester, kodér – člen tímu vykonáva v podstate všetky tieto funkcie. Má teda možnosť, ale zároveň aj povinnosť, výrazne ovplyvniť podobu a fungovanie vyvíjaného software. Jednotliví členovia tímu nemajú pridelenú prácu, ale sami si vyberajú, na čom budú pracovať, zo zoznamu úloh dohodnutých so zákazníkom. V prípade ťažších úloh sa na riešení môžu podieľať viacerí členovia tímu. [13]

### **1.9.8.7 Vývojový tím**

Dôležitou vecou je otázka, ako zložiť vývojový tím tak, aby všetko fungovalo ako má. Cieľom je vytvoriť tím, kde nebudú všetci na rovnakej úrovni v niektorých smeroch, ale tím kde budú pokryté všetky odbornosti, ktoré potrebujeme pri vývoji. Zjednodušene povedané, je potrebný taký tím, ktorý bude schopný riešiť komplexné úlohy každej fázy životného cyklu vývoja aplikácie. Druhou dôležitou vecou je ľudská všestrannosť, kde záleží najmä na ľudských vlastnostiach, medzi ktoré určite patria skúsenosti a seniorita pracovníkov. Ale zloženie tímu len zo seniorov tiež nie je vhodným riešením. Naznačujú tomu hlavne tieto tri dôvody.

1. Cena – plat seniora bude ďaleko vyšší než plat mladého absolventa. A je potrebné vziať do úvahy, že nie každá úloha si vyžaduje špičkového experta s bohatými skúsenosťami.
2. Frustrácia – frustrácia skúseného člena tímu riešiť menšie a ľahšie úlohy, ktoré by zvládol aj menej skúsený absolvent.
3. Rivalita – pri dvoch skúsených vývojároch sa môže stať, že nebudú chcieť spolupracovať a každý si bude chcieť presadiť svoju pravdu. [12]

Je potrebné ešte spomenúť, že optimálna veľkosť vývojového tímu, by mala byť od troch do deväť členov. Stanovenie optimálneho počtu členov je kľúčové pri tom, aby bol tím schopný urobiť požadovaný objem práce.[18]

### **1.9.8.8 Zákazník**

Vo všeobecnosti sú agilné procesy iné v tom, že sa snažia zapojiť zákazníka do projektu, aby sám určoval na čom sa bude pracovať, a ktoré úlohy budú mať prioritu. Dôležité je začleniť zákazníka do tímu tak, aby mal pri vývoji silné slovo čo sa týka zmien pri vývoji a návrhu nových funkcionalít. Zákazníkom môže, byť aj človek z firmy, ale aj mimo nej. Dôležitou úlohou je urobiť zo zákazníka partnera pre vývojový tím, k čomu je potrebné splniť niekoľko aspektov ako poznať seba, svoje schopnosti, možnosti a na druhej strane poznať zákazníka, jeho potreby a biznis. Tak isto je dôležitá vzájomná dôvera a rešpekt. [12]

Existujú dve možnosti definície zákazníka:

- Ten kto software objednáva a platí – väčšinou ide o fyzickú, právnickú osobu alebo firmu, ktorá je uvedená ako odberateľ na zmluvách. Často sa nazýva aj právny zákazník.
- Ten kto software bude používať a bude z neho mať úžitok – tu sa jedná o konkrétne osoby, ktoré budú software využívať a priamo s ním pracovať a ten im bude prinášať úžitok. Nazýva sa aj faktický zákazník.[12]

### 1.9.8.9 User Story

User story je užívateľský popis toho, čo by mal systém robiť, a nie ako by to mal robiť. Cieľom je na základe jednoduchých viet definovať jednotlivé scenáre, ktoré užívateľ bude vo výslednom systéme vykonávať. [11] Jednotlivé User Story majú zadané istú podobu ako by mali vyzeráť a tou je:

**AKO <rola> CHCEM <cieľ> tak, aby <úžitok>**

Teda táto základná podoba má tri základné časti a tými sú:

1. Definícia role – v každej role, v ktorej sa osoba môže vo vyvíjanom software nachádzať, má určité úlohy a z toho vyplývajú jednotlivé užívateľské príbehy.
2. Definícia cieľa – definovanie toho, čo chceme so systémom po jeho dokončení vykonávať. Popis by mal byť vecný a jednoduchý na pochopenie.
3. Definícia úžitku – nepovinná časť User Story, hovorí o tom, čo chceme v konečnom dôsledku dosiahnuť.[11]

### 1.9.8.10 Backlog

Pojem Backlog sa dá najlepšie vysvetliť ako zoznam požadovaných úloh, ktoré bude nutné splniť. V kontexte Scrumu je úloha chápaná ako User Story. Správne je teda Backlog zoznam User Stories, ktoré je nutné implementovať so systémom. Rozoznávajú sa dva typy Backlogu:

1. Produktový Backlog – je celkový zoznam užívateľských príbehov. Jedná sa o zoznam úloh, ktoré úplne špecifikujú vyvíjaný software. Najčastejšie je evidovaný ako zoznam alebo tabuľka.
2. Sprint Backlog - je podmnožinou Product Backlogu, a je to zoznam užívateľských príbehov, ktorými sa v Sprinte budeme zaoberať. [11]

### 1.9.8.11 Sprint

Sprint je iterácia, teda jeden z mnohých opakujúcich sa cyklov vo vývoji software. Cieľom každého sprintu je vytvoriť spustiteľnú aplikáciu, ktorá bude validovateľná a testovateľná. Takže plánovanie každého sprintu by malo byť vždy robené tak, aby jeho výsledkom bol spustiteľný software. [11]

Sprint má v priebehu vývoja pevnú dĺžku a nový Sprint sa začína hneď potom, čo sa skončí predchádzajúci. Pri nastavovaní vhodnej dĺžky Sprintu je nutné brať ohľad na de veci, ktorými sú:

- Schopnosť riadiť Sprint
- Schopnosť v rámci Sprintu vykonať užitočnú prácu [11]

Optimálna dĺžka Sprintu je od dvoch do štyroch týždňov a akceptovateľná dĺžka Sprintu je od jedného do siedmich týždňov. V nasledujúcej tabuľke budú predstavené parametre, ktorými sa dá riadiť pri zvolení dĺžky Sprintu.

**Tabuľka 4: Nastavenie dĺžky Sprintu** (Zdroj: vlastné spracovanie podľa [11])

Kratšie Srinty	Dlhšie Srinty
<ul style="list-style-type: none"><li>• Rýchlejší obrat projektu</li><li>• Častejšie ukážky zákazníkovi</li><li>• Požiadavky sa často menia</li><li>• Nestabilná výkonnosť tímu</li><li>• Prednosť častejšiemu plánovaniu</li></ul>	<ul style="list-style-type: none"><li>• Väčší postup v jednom sprinte</li><li>• Stabilné požiadavky</li><li>• Stabilná a ľahko nastaviteľná výkonnosť tímu</li><li>• Prednosť väčšiemu objemu práce</li></ul>

Pri nastavovaní dĺžky Sprintu je potrebné sa zamýšľať nad filozofiou vývoja, nad tým, čo bude lepšie pre zákazníka a vývojový tím:

- Je žiadané dodávať väčšie a komplexnejšie celky, alebo menšie časti spustiteľného celku častejšie.

- Je žiadané plánovať častejšie a potom pracovať dlhšie, alebo dostávať odozvu od zákazníka častejšie a mať možnosť vykonávať korekcie práce. [11]

#### **1.9.8.12 Meetingy v Scrum**

Vývojový cyklus v Scrum obsahuje niekoľko meetingov, ktoré sú základom tejto agilnej metodiky. Tieto meetingy úzko spolupracujú so základnými artefaktami, ktoré už boli zadefinované a vytvárajú príležitosť pre kontrolu a adaptáciu procesu ako takého. [13] Meetingy sa členia do troch základných kategórií:

1. Plánovacie – väčšinou sa snažia nejakú aktivitu naplánovať, môže sa jednať o plánovanie jedného Sprintu alebo celého projektu. Výsledkom týchto meetingov by mal byť nejaký logický plán.
2. Hodnotiace - tieto meetingy vyhodnocujú, či všetko prebehlo tak, ako bolo naplánované na plánovacom meetingu. Tieto meetingy môžu byť aj viac vyhrotené ako plánovacie, pretože nie vždy ide všetko tak ako má, a niekedy musí dôjsť k priznávaniu chýb jednotlivými členmi.
3. Hodnotiace a Plánovacie – tento meeting je špecifický, pretože je len jeden a volá sa Daily Meeting. Hodnotí sa urobená práca za minulý deň a plánuje sa čo sa bude robiť aktuálny deň. [11]

V nasledujúcej časti budú zadefinované základné meetingy, ktoré sú neoddeliteľnou súčasťou Scrumu.

#### **Plánovací meeting**

Plánovací meeting je jeden z najdôležitejších meetingov Scrumu, a rozhoduje o tom aká bude úspešnosť nadchádzajúceho Sprintu. Základom je pripraviť nasledujúci plán Sprintu, a ten sa skladá z nasledujúcich krokov:

- Definovanie cieľov sprintu.
- Vytvorenie Sprint Backlogu.
- Definícia potrieb pre ďalšie jednania. [11]

Na plánovaní by už malo byť jasné ohodnotenie jednotlivých User Stories, ktoré sa väčšinou odohráva na meetingoch pred plánovaním, ktoré sú zamerané na rozoberanie

problémov do User Stories a ich ohodnotenie. Podobne by mali určite byť stanovené aj priority jednotlivých User Stories.[12]

Stanovenie cieľa je teda prvým základným bodom plánovacieho meetingu. Na základe definovaného cieľa by malo nasledovať podrobnejšie stanovenie plánu z User Stories, ich implementácia, výsledkom ktorej, by mal byť spustiteľný program, ktorý bude testovateľný.[11]

### **Scrum Meeting**

Je časovo ohraničený meeting, ktorý by mal trvať do 15 minút, kde jednotlivý vývojári zdieľajú svoje aktivity od posledného Scrum Meetingu, a podobne ich plán do nasledujúceho Scrum Meetingu. Dobrým zvykom je udržiavať ten istý čas meetingu každý deň, aby si každý člen tímu čo najskôr zvykol a prispôbil sa mu.[12]

Úlohou každého člena tímu je zodpovedať tieto tri základné otázky:

1. Čo som dokončil včera?
2. Čo dokončím dnes?
3. Problémy? [12]

Výsledkom tohto meetingu by mal byť aktuálny stav jednotlivých členov tímu na daných úlohách. V prípade, že to niekde viazne, má tím dostatok času nejako zareagovať, tak aby sa problémová úloha do konca Sprintu stihla.[12]

### **Sprint Retrospective Meeting**

Tento meeting je súčasťou aj niektorých neagilných metód. Jedná sa o efektívny nástroj na získanie spätnej väzby, ktorá môže byť použitá na zlepšovanie a inovovanie ako celých tímov, tak aj jednotlivcov.

Retrospektíva má tieto fázy:

1. Úvod – pripomenutie pravidiel retrospektívy a zoznam akcií z minulého retra.
2. Zber dát – nazbieranie informácií z aktuálneho stavu (čo funguje a čo by sa dalo zlepšiť).
3. Hlbšie porozumenie informáciám – identifikovanie príčiny a pochopenie podstaty problému

4. Brainstorming možností – je základom retrospektívy. Je nutné vytvoriť zoznam akcií, ktorými by mali jednotlivé problémy odstrániť alebo zlepšiť.
5. Zhrnutie konkrétnych akcií – konečným krokom je zjednotenie akcií, ktoré je potrebné vykonať tak, aby sa situácia zlepšila. [12]

Tento meeting slúži na vyhodnotenie predchádzajúceho Sprintu ako takého. Jednotlivý členovia prezentujú svoje návrhy a pripomienky, ktorých výsledkom by mali byť samotné akcie na zlepšenie. Scrum Master vystupuje v úlohe moderátora. [13]

Existujú rôzne typy a modely retrospektív, ktoré sa dajú použiť a aplikovať, avšak je nutné aby retrospektíva obsahovala spomenuté fázy a pravidlá, inak bude strácať význam.[13]

### **Sprint Review**

Sprint Review sa koná vždy na konci Sprintu za účelom kontroly implementácie, ktorá prebehla počas Sprintu. Súčasťou tohto meetingu je prezentovanie implementácie jednotlivých User Stories zákazníkom. Ten si môže novú funkcionálnosť vyskúšať a prezentovať svoje pripomienky alebo zlepšenia, ktoré môžu byť zahrnuté do ďalších Sprintov. [12]

Pravidlom je, že prezentovať by sa mali plne ukončené a uzatvorené User Stories, ktoré sú považované za úplne hotové. Na Sprint Review by mal byť prítomný celý Scrum Tím spolu so zákazníkom. [12]

Tieto meetingy patria medzi základné kamene Scrumu, a bez nich by Scrum len ťažko fungoval. Okrem definovaných meetingov však existujú aj ďalšie, ktoré nemusia byť také pravidelné, ako boli tie, ktoré boli doteraz spomenuté. V prípade ďalších meetingov sa môže jednať o zahájenie a ukončenie projektu, prípadne pred-plánovanie alebo ohodnocovanie User Stories Produktového Backlogu. [11]



## 2 ANALÝZA SÚČASNEJ SITUÁCIE

Táto kapitola bude venovaná predstaveniu a analýze spoločnosti, ktorú som si vybral. Pri analýze budú využité nástroje, ktoré boli zadané a vysvetlené v predchádzajúcej kapitole. Cieľom tejto kapitoly je zoznámiť čitateľa s aktuálnou situáciou a procesmi, ktoré sú vo firme nastavené.

K zhodnoteniu aktuálneho stavu v spoločnosti je potrebné na začiatku urobiť kritickú analýzu momentálnej situácie, k čomu použijeme analýzy interných faktorov, odvetvového okolia, všeobecného okolia a podobne aj SWOT analýzu. Nakoľko výsledkom tejto práce by mala byť zmena vo firme, tieto analýzy boli nutným počiatočným základom, na ktorom by sme zmeny mohli postaviť. V nasledujúcich podkapitolách si jednotlivé analýzy bližšie rozoberieme a pomocou nich si priblížime aktuálnu situáciu v spoločnosti z rôznych pohľadov.

### 2.1 Predstavenie spoločnosti

V tejto diplomovej práci bude riešená problematika úzko spätá s firmou Ixperta, s hlavným sídlom v Prahe. Ostatné pobočky sa nachádzajú naprieč Českou ale aj Slovenskou republikou a to v mestách ako Praha, Brno, Ostrava, Bratislava a Mladá Boleslav. Skupina Ixperta združuje viac spoločností ako sú Ixperta s.r.o, Ixperta Solutions s.r.o, Ixperta International s.r.o a Ixprojekta s.r.o.

**Tabuľka 5: Základné informácie o firme iXperta s.r.o** (Zdroj: vlastné spracovanie)

Názov firmy	iXperta s.r.o
Právna forma	Spoločnosť s ručením obmedzeným
Identifikačné číslo	27599523
Sídlo	Lihovarská 1060/12, Libeň (Praha 9), 190 00 Praha
Rok založenia	13.9.2006
Základný kapitál	56,4 miliónov Kč.

Firma Ixperta pôsobí na trhu informačných a komunikačných technológií cez 25 rokov. V rámci tímov spoločnosti Ixperta pracuje okolo 210 expertov. Z historického hľadiska vznikla firma ako súčasť koncernu Siemens. V rámci projektov, do ktorých sa

spoločnosť zapája, je väčšina projektov zameraných v oblasti komunikácií, infraštruktúry, bezpečnosti a software. Spoločnosť disponuje certifikáciami od predných svetových výrobcov technológií a disponuje tak isto aj vlastným vývojovým software.



Obrázok 10: Logo Ixperta (Zdroj: [22])

### 2.1.1 Zameranie spoločnosti

Zákazníci spoločnosti sa prevažne nachádzajú v Českej Republike, na Slovensku, v Európe, v Ázii a v Severnej Amerike, a to vo všetkých typoch organizácií, od komerčných firiem a výrobných podnikov až po štátnu správu a finančný sektor. Spoločnosť sa sústreďuje hlavne na projekty v oblasti kontaktných centier, dispečerských pracovísk, bankových dealingov, pracovísk operátorov, tiesňových komunikačných systémov, nahrávacích a monitorovacích zariadení, telekonferencií a videokonferencií. Skrátene by sa dalo povedať, že spoločnosť sa zaoberá najmä navrhovaním, vyvíjaním, implementáciou, udržovaním a niekedy tak isto aj samotnou prevádzkou systémov na prenos hlasu, videa a dát.

V rámci projektov firma podporuje štandardy ISO, ITIL, Unify, Cisco, Asterisk a mnoho ďalších, čo pre zákazníka rozhodne predstavuje zaručenú kvalitu. Hlavnými zákazníkmi v oblasti komunikačnej infraštruktúry sú rozhodne Škoda Auto, Ministerstvo práce a sociálnych vecí, kde sa jedná o najväčšie centrálné IP komunikačné centrum v českej štátnej správe.

V oblasti vývoja software sú to hlavne logistické, výrobné a obchodné systémy, podpora obchodu, reporting business dát, embedded systém vrátane hardware,

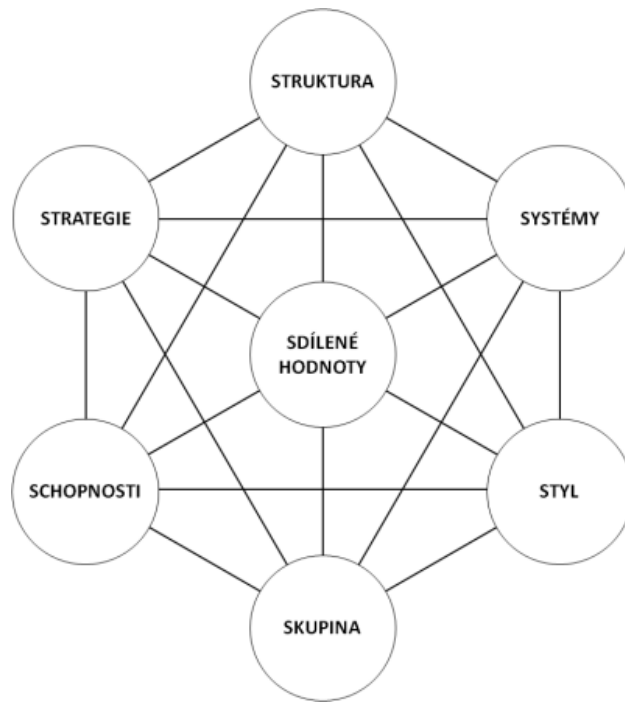
spracovanie audia a videa a tiež signálové procesory. V tejto oblasti je na vysokej úrovni postavená aj podpora, ktorá funguje v podstate nepretržite a je garantovaná v definovanej kvalite (SLA).

### **2.1.2 Smer rozvoja spoločnosti v budúcnosti**

Dá sa povedať, že zo spoločnosti, ktorá sa v minulosti zameriavala hlavne na technológie komunikačných systémov a telefónnych ústrední, sa v súčasnosti stáva firma, ktorá sa začína venovať kompletnej IT infraštruktúre, IT bezpečnosti a vývoju software. V poslednej dobe sa spoločnosť začína orientovať na systémy IoT, autonómnu dopravu a automatizáciu v priemysle, kde Ixperta vyvíja produkt orientovaný na optimalizáciu kontroly kvality vo výrobe, vyvíjaný pomocou strojového učenia, neurónových sietí a umelej inteligencie. Súčasným trendom je automatizácia a robotizácia, ktoré môžeme spolu nazvať ako Priemysel 4.0 a práve týmto smerom by sa chcela spoločnosť uberať a získavať nové projekty.

## **2.2 Analýza interných faktorov (Rozbor 7S faktorov)**

Úlohou tejto analýzy je zdefinovanie a popísanie siedmich základných interných faktorov, ktoré sa sústreďujú na vnútorné prostredie podniku. Jednotlivé faktory sú medzi sebou prepojené a majú základný vplyv na úspešnosť firmy. Medzi jednotlivé faktory, ktoré sú v rámci tejto časti analyzované sú stratégia, štruktúra, systém riadenia, štýl riadenia, spolupracovníci, zdieľané hodnoty a schopnosti.



**Obrázok 11: Analýza interných faktorov** (Zdroj: [14])

## Stratégia

Stratégia danej firmy je určená dlhodobou orientáciou firmy, dlhodobým zamýšľaním nad tým, kam a k akému cieľu bude firma napredovať, a aké má v tomto prostredí možnosti, ktoré umožňujú tieto ciele uskutočňovať. Mala by byť definovaná víziou firmy (ako si to majiteľ alebo vedenie predstavuje) a z konkrétneho poslania firmy.[14]

Na základe tejto definície sa dá povedať, že hlavným cieľom je do budúcnosti určité udržanie základných projektov, ktoré firme zabezpečujú hlavný zdroj príjmov a to hlavne dodržiavanie zmluvy s hlavnými zákazníkmi. To je základným predpokladom aj pre vytvorenie nových projektov aj v smeroch, v ktorých zatiaľ firma zatiaľ nepôsobila. Zároveň by to vytvorilo aj istotu pre spoločnosť do budúcnosti, aby by nebola závislá len na financiách od zákazníkov, s ktorými spolupracuje dlhú dobu. Je dôležité spomenúť, že vedomosti, ktoré firma má v oblasti IP telefónie a problematiky s ňou spojenou, je možné vhodne využiť aj v rámci rozvoja v nových projektoch.



Obrázok 12: OpenScape Desk Phone CP600 (Zdroj : [21])

## Štruktúra

Nakoľko má firma pomerne veľa zamestnancov, tak je štruktúra firmy členená podľa projektov, na ktorých zamestnanci pracujú. Čo sa týka hlavného produktu tak v iXperte sú tímy, ktoré sa sústreďujú na istú časť produktu pričom úzko spolupracujú s kolegami v iných krajinách, ako je napríklad Turecko, Anglicko, Nemecko alebo Grécko. Vývoj v Brne je orientovaný na IP telefóniu, kde sa tímy sústreďujú na implementáciu a podporu koncových zariadení, ale taktiež aj ústrední. Riadenia vývoja a udržiavanie produktu pre zákazníkov je svojou štruktúrou pomerne náročné a komplikované. Ako už bolo predtým spomínané, tak iXperta sa snaží investovať aj do nových projektov v iných smeroch ako sú komunikačné technológie. Tým pádom vznikli tímy, ktoré robia na rozdielnych produktoch. Formálne by sa dalo povedať, že táto štruktúra sa podobá líniovo štábnej organizačnej štruktúre.

## Systém riadenia (Informačné systémy)

Systém riadenia je pomerne zastaralý a to vytvára problémy pri riadení projektov, riešení problémov a tak isto pri ich archivácii a dohľadávaní. Pri udržiavaní a následnej dostupnosti informácií pri tiketoch sa používa interný nástroj MRTS, ktorý je využívaný naprieč všetkými projektami. Na ostatné informácie, ako sú dokumentácie, postupy, návrhy pri vývoji sa využíva ďalší interný nástroj, ktorý je podobne už starší a vyžaduje si modernizáciu a vylepšenia. Okrem toho sú dôležité informácie a dokumenty ukladané aj na rôznych zdieľaných úložiskách, s prístupom pre každého, kto ich potrebuje.

V tomto smere si firma vyžaduje modernizovanie a viaceré vylepšenia. Hlavne sa to týka zavedenia nového informačného systému, pomocou ktorého bude archivovanie a dohľadávanie informácií jednoduchšie a intuitívnejšie. Z procesného hľadiska by bolo vhodné mať systém, ktorý bude lepšie zobrazovať prácu na tiketoch a ukazovať aktuálny stav v akom sa nachádzajú.

### **Štýl riadenia**

Čo sa týka rozhodovania, tak tu môžeme skonštatovať, že ide o autoritatívny štýl riadenia, kde samotní zamestnanci nemajú slovo pri riadení firmy a nemajú možnosť výrazne ovplyvňovať rozhodnutia. V súčasnosti funguje systém, kde majú manažéri hlavné slovo pri konečných rozhodnutiach, a tak isto je firma limitovaná tým, že zákazník schvaľuje projekty a novú funkcionality, ktorá sa bude v nasledujúcom období vyvíjať.

Víziou do budúcnosti je určite dosiahnutie demokratického štýlu riadenia, až liberálneho štýlu riadenia, hlavne na nových projektoch, kde by samotní zamestnanci mali isté slovo pri rozhodnutiach. Práve od ich zapojenia si firma sľubuje isté výhody v rámci získavania nových projektov. Preto je zmena štýlu riadenia jedným z hlavných cieľov firmy v blízkej dobe. Kľúčovú rolu by mali mať pri rozhodovaní o vývoji produktov aj zákazníci, a preto bude zmena štýlu a celého prístupu riadenia žiadaná.

### **Spolupracovníci**

Zamestnanci firmy sú naozaj dôležitou súčasťou firmy, na ktorých je firma postavená, nakoľko sa jedná o odborníkov, ktorí stáli pri základných projektoch hlavne v čase, keď bola spoločnosť súčasťou koncernu Siemens. Ixperta sa snaží a určite by mala stavať na svojich zamestnancoch a vytvárať im vhodné podmienky do budúcnosti. Ale aj tu sa určite nachádzajú miesta kde má firma rezervy, a ktoré by sa dali zlepšiť v porovnaní s konkurenciou.

Celkovo môžeme ľudí vo firme považovať za silnú stránku, čo sa týka konkurencieschopnosti na trhu, ktorá je výsledkom firemnej kultúry, motivačného zázemia a kvalitných podmienok, ktoré firma zamestnancom poskytuje.

### **Zdieľané hodnoty**

Dôležitou zdieľanou hodnotou je udržiavanie firemnej kultúry, ktorá má podstatný podiel na úspešnosti firmy. Vo firme by mali byť zdieľané predstavy a ciele na všetkých

pozíciách a zamestnanci by mali byť o nich presvedčení. Kultúra firmy charakterizuje aj vnútornú atmosféru firmy, takže sa dá povedať, že sa jedná o určitý nehmotný produkt, kde je výsledkom myslenie ľudí vo firme a činnostiach, ktoré sa vo firme robia.

### **Schopnosti**

Tu by sa dalo povedať, že firma má produkty, kde nemá konkurentov na trhu, alebo ich má veľmi málo. V rámci produktov v oblasti komunikačných technológií, ktoré firma vytvára, existuje jeden hlavný konkurent, s ktorým firma súperí, ale jej výhodou sú bohaté a dlhoročné skúsenosti od jej vzniku. Môže sa oprieť o množstvo špecialistov so skúsenosťami v jednotlivých odboroch a rovnako aj pri podpore u novších projektov.

## **2.3 Analýza okolitého prostredia organizácie (SLEPT analýza)**

### **Sociálne faktory**

Čo sa týka celkového záujmu v odvetví komunikačných a informačných technológií, môžeme pozorovať obrovský nárast oproti situácií pred desiatimi rokmi. Podobný trend môžeme očakávať aj naďalej, aj keď nie s takým rapídny rastom. To zapríčiňuje aj nedostatok zamestnancov v tomto odvetví, pričom môžeme očakávať, že dopyt po zamestnancoch bude aj naďalej pretrvávať. Konkurencia medzi firmami narastá, a to má za následok, že spoločnosti musia svoje ponuky stále zlepšovať a vytvárať zamestnancom výhodnejšie podmienky. Tu je určite priestor na rozvoj a získanie nových špecialistov pre jednotlivé projekty.

### **Legislatívne faktory**

Z tohto hľadiska je pre firmu dôležité dodržiavanie právnych a legislatívnych noriem v oblasti podnikania, kde radíme najmä obchodné právo a daňové zákony. Nakoľko firma podporuje rôzne certifikácie a normy je nutné, aby ich pravidlá striktne dodržiavala. Kľúčové je najmä dodržiavanie noriem v oblasti bezpečnosti, pretože tá je jednou z najzraniteľnejších v dnešnej dobe informačných technológií.

### **Ekonomické faktory**

Medzi faktory ovplyvňujúce podnikanie patria napr. menová stabilita alebo miera inflácie. V podnikaní má výška nezamestnanosti tiež vysoký vplyv. Nezamestnanosť mala v posledných rokoch klesajúcu tendenciu a dnes je na hodnote 2,3%, teda je pomerne nízka. Hlavne v oblasti informačných technológií je veľký dopyt po

odborníkoch v tomto odvetví, čím vzniká u zamestnávateľov povinnosť zlepšovať podmienky, ak chcú mať vo svojej firme najlepších ľudí na trhu.

### **Politické faktory**

Momentálne je politická situácia v Českej republike stabilizovaná, je pevnou súčasťou Európskej únie a očakáva sa, že tento trend bude aj naďalej pokračovať. Napriek tomu, každá zmena vlády a obsadenia parlamentu vytvára tlak na zmeny a tie majú za následok najmä zmeny v legislatívnej oblasti. V prípade väčších zmien by to mohlo mať vplyv aj na podnikanie spoločností v IT sektore. Čo sa týka spoločnosti Ixperta a jej hlavného zákazníka, spolupracuje s pobočkami aj v iných štátoch, čím je závislá aj na politickej situácii najmä v EÚ ako aj mimo nej. Práve zmeny, ktoré v dnešnej dobe hrozia v rámci Európskej únie napríklad v súvislosti s odchodom Veľkej Británie, by mohli v značnej miere ovplyvniť podnikateľské prostredie.

### **Technologické faktory**

Je to jeden z faktorov, ktorý má v tomto odvetví pomerne veľký, ak nie najväčší vplyv. Technológie majú v súčasnosti obrovský vplyv na spoločnosť a preto sú dôležité aj v podnikaní. Je dôležité, aby firma držala krok s novými technológiami a stále sa vyvíjala. Iba tak si môže upevniť svoje postavenie a zabezpečiť si svoju existenciu na trhu.

## **2.4 Porterova analýza piatich síl**

Porterov model je dôležitý pri analýze podnikateľského prostredia v blízkosti okolia firmy. Poskytuje informácie o súčasnej pozícii firmy na trhu. Tento model je vhodné používať aj pre objavovanie nových produktov (substitútov), pretože smerovanie a sila zákazníkov a odberateľov by sa mala stať hybnou silou k novým strategickým cieľom.

### **Konkurencia prostredia**

V rámci IT odvetvia je momentálne konkurencia na trhu obrovská a môžeme predpokladať aj narastajúcu tendenciu. V súčasnosti je veľmi ťažké nájsť nejakú dieru na trhu, prísť s niečím úplne novým a uspieť tak výrazne u zákazníkov. Priamym konkurentom na trhu je určite firma Cisco, ktorá pôsobí na trhu takisto dlho. Momentálne sa nepredpokladá zvýšenie počtu konkurentov vo výrobe desktopových telefónov nakoľko momentálne je doba mobilných telefónov, ktoré prevažujú. Napriek



tomu má firma stabilných zákazníkov, s ktorými jednoznačne počíta aj do budúcnosti. Čo sa týka produktov v iných oblastiach informačných technológií, tak existuje viac konkurentov, s ktorými firma súperí na trhu a dá sa predpokladať ich nárast.

### **Hrozba substitútov**

Tu by sa dalo povedať, že hlavným substitútom pre kancelárske telefóny sú za posledných 10 rokov určite mobilné telefóny, ktorých trend začal vo väčšom niekedy na začiatku tohto tisícročia. Vývoj v oblasti mobilných telefónov je neustále obrovský a pravdepodobne bude naďalej pokračovať. Pri kancelárskych telefónoch a ústredniach, s ktorými telefóny komunikujú sa nedá očakávať tak enormný vývoj do budúcnosti.

### **Zákazníci**

Zákazníkmi sú v prípade iXperty väčšie firmy, ktoré využívajú kancelárske telefóny hlavne v rámci komunikácie či už medzi zamestnancami alebo aj v rámci mimo firemných volaní. Medzi najvýznamnejších zákazníkov patrí napríklad Ministerstvo práce a sociálnych vecí v Českej republike a tak isto Škoda Auto a veľa ďalších. Dá sa povedať, že dopyt po týchto telefónoch je stále dostatočne veľký a stále existujú stabilní zákazníci aj napriek tomu, že v dnešnej dobe používa každý mobilné telefóny. Napriek tomu práve požiadavky zákazníkov budú zohrávať dôležitú úlohu pri vývoji produktov a ich udržiavaní do budúcnosti.

### **Dodávatelia**

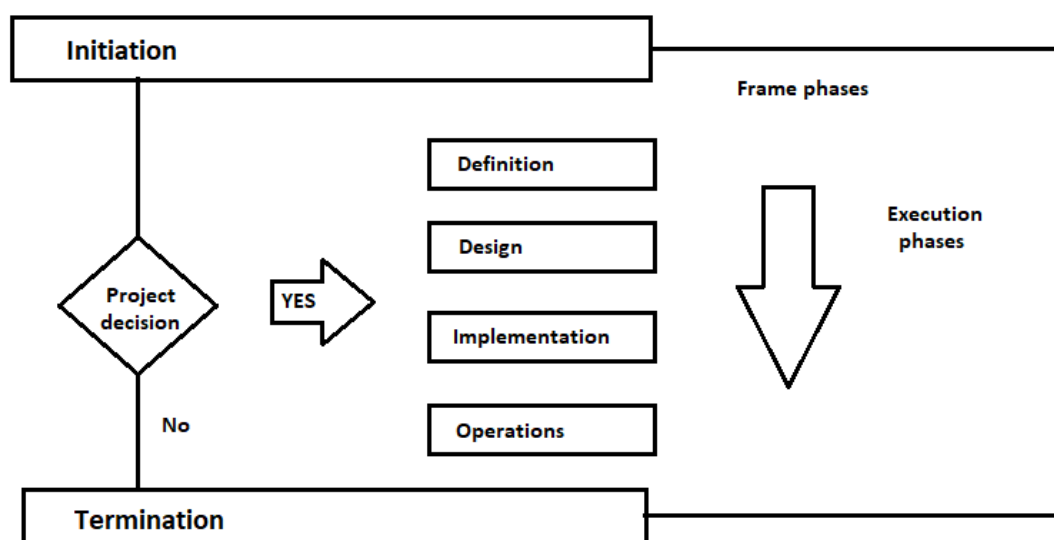
Na zoznam dodávateľov by sa dal zaradiť určite celý hardware, ktorý sa ako taký dodáva do iXperty a práve pre tento hardware je potom vyvíjaný software. To znamená, že pri telefónoch sa jedná o všetky časti od základnej dosky až po obal telefónu. To isté platí aj pre ústredne, s ktorými telefóny komunikujú. Dovoz hardware je preto pre firmu dôležitý a každá zmena dodávateľa môže mať za následok výraznú zmenu aj v softvérovej časti, na ktorej pracujú vývojové tímy v iXperte.

### **Riziko vstupu nových konkurentov**

Na momentálnom trhu je zvyšovanie počtu konkurentov v oblasti IT vysoké, ale keď budeme brať do úvahy projekt s vývojom software pre desktopové telefóny, tak tu nárast nepredpokladáme. V prípade, že by sa firma chcela v budúcnosti orientovať v inom smere informačných technológií, tak je nutné očakávať pomerne veľkú konkurenciu, s ktorou sa bude musieť firma vysporiadať.

## 2.5 Analýza priebehu vývoja vo firme

V tejto podkapitole bude priblížený priebeh procesu vývoja vo firme v jednotlivých tímoch. Doterajší priebeh bol založený na tradičných metódach, ktoré boli spomenuté v predošlých podkapitolách. Priebeh, ktorý je popísaný v tradičných metódach bol upravený tak, aby bol pre firmu použiteľnejší a efektívnejší.



Obrázok 13: Proces vývoja projektu (Zdroj: Vlastné spracovanie)

Na predošlom obrázku sú zobrazené jednotlivé fázy priebehu projektov tak ako na seba logicky nadväzujú. V počiatočnom stave sa rozhoduje či sa daný projekt oplatí riešiť alebo nie. Táto fáza zahŕňa aj pomerne intenzívnu komunikáciu so zákazníkom a slúži na získanie podstatnej časti informácií o projekte. V tomto období je potrebné naplánovanie priebehu projektu s podrobnejšou analýzou a stanovením termínov. Tak isto je dôležité stanovenie podmienok medzi zákazníkom a spoločnosťou a spísanie potrebných zmlúv. Táto fáza sa nazýva aj „project kick off“ a jej výsledkom je podrobná špecifikácia budúceho riešenia ktorá obsahuje dohody medzi zákazníkom a spoločnosťou a tak isto je stanovený plán vývoja projektu.

V prípade, že sa spoločnosť so zákazníkom dohodnú, že na projekte spolupracovať nebudú, prechádza sa do stavu termination, kde je výstupom výsledný report o projekte s ďalšou dokumentáciou a dátami, ktoré sa archivujú.

V prípade pokračovania na projekte sa postupne prechádza fázami, ktoré sú spísané v tabuľke.

**Tabuľka 6: Podrobný popis jednotlivých fáz a podfáz** (Zdroj: vlastné spracovanie)

Fáza	Podfáza	Činnosti
Definícia	Definícia požiadaviek	Definícia hlavných požiadaviek pokiaľ neboli definované
	Definícia produktu	Definovanie detailnejších požiadaviek na software na vytvorenie základného modelu
	Nakreslenie základného modelu	Základný návrh na model software
Dizajn	Architektonický dizajn	Prvotný grafický návrh implementácie
	Detailný dizajn	Grafický dizajn s bližšími špecifikáciami na implementáciu
Implementácia	Produkcia zdrojového kódu	Vývoj na základe predošlých návrhov
	Adaptácia vytvoreného software	Zpracovanie vytvoreného SW
	Integrácia a testovanie	Integrácia a vytvorenie testov na vytvorený software
	Príprava na produkciu	Dokončenie dokumentácie a postupu integrácie
Prevádzka	Zavedenie pilotného produktu	Skúšobné zavedenie produktu
	Zavedenie produktu do prevádzky	

### Definičná časť

V definičnej časti je podstatné zdefinovanie a stanovenie hlavného cieľa projektu na základe stanovených požiadaviek od zákazníka a vytvorenie počiatočného prototypu

z definovaných cieľov. Súčasťou tejto fázy je aj zadefinovanie externých rozhraní a podmienok, ktoré má výsledný produkt spĺňať. Výsledkom tejto časti by mali byť užívateľské a softvérové špecifikácie a približný odhad úsilia, ktoré bude potrebné dať na vývoj.

### **Dizajnová časť**

Dizajnová časť obsahuje základnú špecifikáciu požiadaviek od zákazníka spolu s navrhnutím architektonického dizajnu s hlavnými komponentami produktu. Na základe dátových modelov sú vytvorené prototypy a architektonický dizajn celého software. Z hrubo vytvoreného dizajnu sa detailnejšie definujú jednotlivé komponenty a je vytvorený špecifickejší model. Potom sa dotačne navrhujú úpravy potrebné na napojenie novej funkcionality na existujúci software. Výsledkom celej tejto fázy by mal byť kompletný návrh novej časti software, na základe ktorého bude prebiehať implementácia.

### **Implementácia**

Vstupnou časťou implementácie by mali byť pripravené návrhy dizajnovnej časti, podľa ktorých by sa mal produkt vyvíjať. Tak isto sú vybrané aj nástroje a programovací jazyk v ktorom sa bude nová funkcionality vyvíjať. Nepísaným zvykom je aj tvorba testov a dokumentácie v priebehu vývoja. Po tom čo sa vývoj nových komponentov dokončí, prichádza integračná časť, v rámci ktorej sa rieši prispôsobenie nových komponentov k existujúcemu software a rozhraniam. V prípade, že je táto časť dokončená, dochádza k testovaniu software testermi, kde by mali byť zachytené chyby, ktoré vznikli pri implementácii, alebo pri integrácii novej časti software. Poslednou časťou implementačnej fázy je dokončovanie dokumentácie a príprava na predanie produktu zákazníkovi. Výsledkom implementačnej časti by mal byť hotový produkt s dokumentáciou a plánom zavádzania.

### **Zavedenie produktu do prevádzky**

Základom tejto časti je príprava pilotnej produkcie, v rámci ktorej sa produkt dostáva k zákazníkovi za účelom otestovania software v plnom vyťažení. K produkcii pilotného produktu patrí aj podpora a samotná inštalácia software. Ak sa v pilotnej verzii nejaké chyby vyskytnú, vývojárske tímy sa ich snažia opraviť a zahrnúť do konečnej verzie.

Výsledkom tejto fázy by mal byť hotový a odladený produkt, ktorý by mal spĺňať akceptačné kritéria stanovené na začiatku.

Hlavným problémom pri vodopádovom modeli vývoja je, že zákazník nie je začlenený do priebežného vývoja a nevie prípadne reagovať, či sa budúci produkt správa tak, ako bolo po ňom požadované na začiatku. Dôvodom tohto problému môže byť nedorozumenie medzi zákazníkom a vývojármi v definičnej časti. Výsledok tohto nedorozumenia sa neskôr odzrkadlí pri vývoji a konečná funkcionálna sa výrazne líši od tej, ktorú zákazník chcel na začiatku. Druhou príčinou môže byť, že zákazník na začiatku úplne presne nevie špecifikovať, ako si výsledný produkt predstavuje a na konci zistí, že to čo mu bolo dodané mu nevyhovuje. Vo všeobecnosti sú tieto dve chyby najčastejšími príčinami neúspešného projektu a spôsobujú jednak časové ale aj finančné straty či už pre zákazníka alebo pre firmu. Medzi vhodné riešenia tohto problému pripadá zmena procesu vývoja na agilný. To by prinieslo hlavne požadovanú zmenu prístupu zákazníka počas celého procesu vývoja produktu. Agilný prístup zvýrazní možnosť interakcie vývojárov so zákazníkom a prezentovanie čiastočného riešenia. Práve toto sú dôvody, pre ktoré by mala spoločnosť nasadiť agilný prístup.

Ďalším problémom je dôraz na testovanie software počas implementačnej časti a po jej dokončení. Často sa stane, že produkt ide do produkcie s tým, že nie je poriadne otestovaný. Tým vznikajú chyby, kvôli ktorým musia byť robené produkcie s dodatočnými opravami. Zavedenie agilného prístupu by dávalo význam aj z pohľadu testovania a tento proces by mohol byť efektívnejší a nevznikalo by toľko chýb.

## **2.6 SWOT analýza**

Na základe zistených informácií z predošlých analýz bola zostavená SWOT analýza, ktorá pozostáva zo silných a slabých stránok interných faktorov a príležitostí a hrozieb externých faktorov firmy.

**Tabuľka 7: SWOT analýza** (Zdroj: vlastné spracovanie)

	<b>Silné stránky</b>	<b>Slabé stránky</b>
<b>Interné faktory</b>	<p>Kvalitný produkt</p> <p>Stabilný zákazník</p> <p>Experti v oblasti komunikačných technológií</p> <p>Kvalitné know-how v odbore komunikačných technológií</p> <p>Firemná kultúra</p>	<p>Málo efektívny proces vývoja</p> <p>Chýbajúci nový produkt</p> <p>Chýbajúci software na vnútro firemnú spoluprácu</p> <p>Chýbajúci software na riadenie procesu práce</p> <p>Motivácia zamestnancov</p> <p>Zlá a neefektívna komunikácia so zákazníkom</p> <p>Málo návrhov na nové funkcie produktu</p>
	<b>Príležitosti</b>	<b>Hrozby</b>
<b>Externé faktory</b>	<p>Produkt nemá veľkú konkurenciu</p> <p>Rozšírenie počtu tímov vo firme</p> <p>Získanie nových projektov na trhu</p> <p>Využitie interného know-how v rámci nových príležitostí na trhu</p> <p>Rozšírenie know-how v iných smeroch IT</p> <p>Častejšia komunikácia a interakcia so zákazníkmi</p> <p>Nové funkcie a nápady do existujúceho produktu</p> <p>Prezentovanie implementovaných novínok zákazníkom</p>	<p>Rýchly vývoj technológií</p> <p>Dopyt konkurencie po odborníkoch v IT</p> <p>Vstup novej konkurencie</p> <p>Substitúty</p> <p>Klesajúci dopyt po produkte pre zákazníkov</p> <p>Získanie nepresných požiadaviek od zákazníka</p> <p>Zákazník nie je súčasťou vývoja produktu</p> <p>Nezáujem zákazníka</p>

Zo SWOT analýzy vyplýva, že firma disponuje kvalitným zázemím so stabilným produktom a kvalitnými odborníkmi, ale mala by sa porozhliadnuť po nových

projektoch, ktoré prinesú úspech v budúcnosti. Produkt ako taký, má pomerne stabilných zákazníkov a dá sa predpokladať, že dopyt po ňom by v nasledujúcom období nemal rapídne klesnúť. Na udržanie dopytu po produkte bude mať v budúcnosti určite vplyv aj komunikácia so zákazníkmi ohľadom nových funkcií a vylepšení, do ktorých by sa oplatilo investovať a vyvíjať ich, a preto bude nutná zmena v procese vývoja a prístupu k zákazníkom.

Spoločnosť má na trhu dobré meno a to by sa dalo využiť pri vstupe do nových projektov v budúcnosti. Tak isto by sa mala firma sústrediť na nábor nových ľudí a odovzdávanie know-how od skúsenejších zamestnancov. To si žiada aj zmeny vo firme na úrovni tímov a naplno využiť ich potenciál a dať im tak isto možnosť na rozvoj vlastných nápadov či už priamo alebo nepriamo komunikovať aj so zákazníkom, pre ktorého je produkt vyvíjaný. Hlavne nový produkt, a nábor nových zamestnancov budú pre firmu zásadné do budúcnosti, a preto by zmena vedená ku zlepšeniu efektivity a priestoru pre nové projekty bola vítaná.

Ďalším dôležitým krokom by malo byť zavedenie software, ktorý by slúžil na vnútro firemné záležitosti či už dokumentácie k jednotlivým funkciám telefónov a ústrední alebo informácie potrebné pre vývojárov alebo manažment. Výrazným posunom vpred by mala byť aj výmena starého systému so zoznamom tiketov za nový a inovatívnejší, čo by určite zlepšilo a sprehľadnilo celý proces vývoja.

Najkritickejším bodom je však priebeh vývoja projektov, do ktorého zákazník nie je zaangažovaný a výrazný problém nastáva často už v počiatočných fázach projektu. Ale najväčším problémom je skutočnosť, že chyby vzniknuté na začiatku projektu sa na produkte odzrkadlia častokrát až na konci, keď zákazník projekt prevezme a testuje. Práve tieto prípady stoja firmu nemalé, hlavne finančné straty a nespokojnosť sa zvyšuje na oboch stranách. Presne táto príčina by mala byť dôvodom na zmenu týkajúcu sa prístupu k vývoju projektov a komunikácie so zákazníkom, čo by malo pomôcť k získaniu presnejších požiadaviek na výsledný produkt.

### 3 NÁVRH RIEŠENÍ A PRÍNOS NÁVRHOV RIEŠENÍ

Na základe analýz urobených v predchádzajúcej kapitole budú navrhnuté a detailne popísané zmeny na zlepšenie a zefektívnenie procesu vývoja pomocou agilnej metodiky Scrum. Z predošlých analýz bolo zistených niekoľko nevýhod a problémov, ktoré by bolo potrebné vylepšiť alebo úplne zmeniť. V niektorých prípadoch sa jedná o menšie problémy, ale tak isto boli zistené aj zásadné problémy, na ktorých spoločnosť stráca jednak finančné, ako aj časové zdroje.

Súčasťou problémov, ktoré boli zistené prostredníctvom analýz sú:

- Zákazník nie je zapojený do vývoja a nemôže si v priebehu vyskúšať nejaké funkčné riešenie.
- Chýbajúce návrhy na nové funkcie software.
- Výsledný produkt nespĺňa požiadavky zákazníka.
- Chýbajúca komunikácia so zákazníkom, a tak isto aj medzi členmi v tímoch.
- Meniace sa požiadavky v priebehu vývoja.
- Nekonzistentnosť medzi návrhom a výslednými parametrami.
- Veľa chýb zistených neskorým testovaním, až po zaintegrování nového produktu.
- Chýbajúci software na proces vývoja projektov vo firme.
- Zastaralý a neprehľadný firemný portál.

V tejto kapitole budú najskôr vysvetlené základné pojmy a artefakty úzko súvisiace so Scrumom a potom budú použité pri konkrétnych návrhoch pre spoločnosť. Snahou týchto návrhov bude celková zmena procesu vývoja vo firme. Zavedením agilnej metodiky Scrum sa očakáva postupné odstránenie nevýhod a problémov vypísaných vyššie. V tejto kapitole bude postupne popísaná transformácia z vodopádovej metodiky, ktorá sa momentálne pri vývoji používa na agilnú metodiku Scrum.



### 3.1 Návrh nových rolí

Na základe výstupov z analytickej časti bude cieľom tejto kapitoly navrhnutie a aplikovanie zmien v procese vývoja vo firme s použitím agilnej metodiky Scrum. Zároveň bude použitá terminológia týkajúca sa Scrumu spomenutá v teoretickej časti diplomovej práce.

Hlavným problémom v aktuálnom procese je celkový prístup zákazníka pri vyvíjaní nových funkcionalít alebo kompletne nových komponentov do existujúceho software. Doterajší prístup sa ukázal byť málo efektívny a vo viacerých prípadoch nastávali komplikácie pri integrovaní nových častí do existujúceho software, kde neboli splnené požiadavky zákazníka, alebo sa výrazne líšili. Následkom toho boli časové aj finančné straty.

Táto kapitola bude zameraná na nastavenie agilnej metodiky Scrum pri vývoji. Dôležitou časťou bude transformácia tímov na Scrum tímy, jasné pomenovanie a určenie rolí v tímoch. Na toto budú použité pojmy a základné artefakty Scrumu spomenuté v teoretickej časti.

Nakoľko sa jedná o zmenu procesu riadenia a celého prístupu, ktoré sa bude týkať každého, od top managementu až po samotných vývojárov, tak prvou dôležitou vecou, ktorú je potrebné urobiť je definovanie a návrh hlavných rolí v Scrum. Medzi základné role v Scrum patria pozície Scrum Mastra, Vlastníka produktu a samotného člena vývojového tímu tak, ako to už bolo spomínané v teoretickej časti tejto diplomovej práce.

#### 3.1.1 Scrum Master

Spolu s rolou vlastníka produktu sa zaraďuje z pohľadu teórie v Scrum medzi manažérske role. Doteraz plnil manažérsku rolu v tímoch človek, ktorý mal väčšinou rolu projektového manažéra. Na základe teórie Scrumu táto funkcia úplne zaniká a vznikajú pozície Scrum Mastra a Product Ownera.

Scrum Master bude mať vo firme tieto úlohy:

- Udržiavanie pravidiel Scrumu počas vývoja.

- Koordinácia meetingov – tak aby boli vecné, riešili sa tam podstatné veci. Bude vykonávať rolu moderátora meetingov. Bude zabezpečovať nachystanie samotného meetingu (zarezerovanie zasadačky a projektorov, poprípade ďalšie technické požiadavky).
- Riešenie problémov týkajúcich sa tímu – členovia tímu by sa mali starať čisto len o svoju prácu. Problémy ako chýbajúce prístupy do systému, problémy s pripojením na internet, ale aj problémy, ktoré spôsobujú vývojárom „nepohodlie“, ako osvetlenie, klimatizácia alebo nefunkčná chladnička by mal mať na starosti Scrum Master.
- Riešenie problémov komunikácie medzi členmi tímu.
- Motivácia členov tímu a vytváranie dobrej atmosféry v tíme – zabezpečenie benefitov (aj nefinančných) pre členov tímu.
- Mal by vytvárať hranicu medzi zákazníkom a vývojovým tímom – Scrum Master bude udržiavať odstup zákazníka od tímu v prospešnej miere.

Riešením tejto situácie by mohlo, byť angažovanie nových zamestnancov so skúsenosťami Scrum Mastrov, a tak isto urobiť prieskum vo firme, kto by sa na takúto rolu hodil alebo by mal záujem. Vhodnou kombináciou týchto dvoch riešení by sa mohla vytvoriť skupina ľudí vhodných na túto pozíciu, ktorej by boli poskytnuté školenia pre Scrum Mastrov v softskills praktikách. Výhodou začlenenia ľudí, ktorých firma vybrala zo svojich radov medzi skúsených Scrum Mastrov je, že sa môžu vzájomne zdokonaľovať a čerpať rady a skúsenosti od svojich skúsenejších kolegov.

Na základe povinností a hlavných úloh, bude potrebné zamestnať nových ľudí, ktorí budú musieť absolvovať školenia, kde budú vysvetlené základné princípy Scrumu. Na túto pozíciu bude potrebné zamestnať nových ľudí, pretože vo firme je málo zamestnancov, ktorý disponujú schopnosťami na vykonávanie tejto funkcie. Výhodou je, že na pozícii Scrum Mastra nemusí, byť človek, ktorý sa vyzná do programovania alebo vývoja software. To znamená, že aj náklady na zamestnanie ľudí na tejto pozícii nebudú až také vysoké ako pri technikoch.

**Tabuľka 8: Návrh pozície Scrum Master** (Zdroj: vlastné spracovanie)

	<b>Nový zamestnanci</b>	<b>Zamestnanci z firmy</b>
<b>Počet Scrum Mastrov</b>	3	1

Na zaškolenie Scrum Mastrov bude potrebné 3-denné intenzívne školenie, ktorého obsahom bude vysvetlenie základných princípov Scrumu, artefaktov a meetingov. Podstatou školenia však bude zameranie sa na hlavné činnosti a úlohy Scrum Mastra s konkrétnymi modelovanými situáciami. Tieto preškolenia absolvujú či už noví alebo aj zamestnanci firmy vybraní na túto pozíciu. Po tomto školení by mali byť Scrum Mastri schopní správne motivovať, podporovať a chrániť tím.

### **3.1.2 Product Owner**

V porovnaní s doterajším procesom, sa tak isto ako pri roli Scrum Mastra jedná o celkom novú rolu, ktorú definuje agilná metodika Scrum. Nakoľko spoločnosť úzko spolupracuje s ostatnými pobočkami vo svete a zákazníkom je tiež zahraničná firma, rolu Product Ownera bude predstavovať človek, ktorý nebude priamo v každodennom kontakte so Scrum tímom. Táto pozícia bude pomerne úzko spojená s rolou Product Owner Proxy, ktorá bude vysvetlená následne. V podstate Product Owner Proxy bude človekom, ktorý bude s Product Ownerom úzko spolupracovať, a tak isto prevezme zodpovednosť, za niektoré jeho právomoci. Pri nasadzovaní agilných metodík môžu nastať situácie, kde je nutné prispôbenie procesov spoločnostiam, tak aby nevznikali problémy v komunikácii alebo v procesných veciach.

Táto rola so sebou prináša ako aj práva, tak aj povinnosti, ktoré musí človek na tejto pozícii zvládať. Rolu Product Ownera bude predstavovať človek, ktorého hlavnou úlohou bude stanovovať a definovať vízie celého projektu spolu so zákazníkom. Po procesnej stránke bude túto rolu zastávať človek, ktorý bude v úzkom kontakte so zákazníkmi a definovať jednotlivé ciele, ktoré by sa mali dosiahnuť.

Na pozíciu Product Ownera budú najvhodnejší ľudia priamo od zákazníka, ktorý sa nachádza v Nemecku. Nakoľko tento zákazník spolupracuje s firmou dlho, je pomerne dobre oboznámený s produktom a jeho funkciami, čo bude určite výhodou. Napriek tomu že pozícia Product Ownera bude obsadená ľuďmi od zákazníka, bude potrebné zabezpečiť aspoň jedného nového Product Ownera, ktorý bude tak isto fyzicky v Nemecku s ostatnými Product Ownermi.

**Tabuľka 9 : Návrh pozície Product Owner** (Zdroj: vlastné spracovanie)

	<b>Nový zamestnanci</b>	<b>Aktuálny zamestnanci zákazníka</b>
<b>Počet Product Ownerov</b>	1	3

Na túto funkciu bude treba pribrať jedného nového človeka, ktorý sa však bude musieť aspoň minimálne stotožniť s produktom a jeho funkciami. Či už pre ľudí, ktorí už s firmou pracovali alebo nových ľudí, ktorí budú zastávať túto funkciu, je potrebné aby firma zabezpečila preškolenie týchto zamestnancov, kde budú vysvetlené princípy Scrumu. Obzvlášť pre Product ownerov budú tieto školenia prínosné, pretože pochopia akým štýlom by mali pracovať, aby bol výstup čo najefektívnejší jednak pre zákazníkov ale aj pre celý vývojový tím.

Obsahom týchto školení by malo byť hlavne zostavovanie Backlogu a samotných User Stories v podobe nových funkcií telefónu s následným určením priorít. Podobne budú využité modelové situácie, na ktorých budú vysvetlené kompetencie Product Ownera, pristupovanie k jednotlivým povinnostiam a motivovanie tímu. Bude sa taktiež jednať o intenzívne 3-dňové školenie.

Po absolvovaných školeniach by človek v tejto pozícii vo firme iXperta mal byť schopný niesť zodpovednosť a mať na starosti:

- Komunikácia so zákazníkom a tvorbu Backlogu.
- Určenie priorít jednotlivým úlohám.

- Udržiavanie produktu v rámci požiadaviek zákazníka - bude zodpovedať za to ako bude projekt v budúcnosti vyzerat'.

### 3.1.3 Product Owner Proxy

Tak ako už bolo spomínané, Product Owner nebude v každodennom kontakte s vývojovým tímom, nakoľko sa fyzicky bude nachádzať v inom štáte, a tak zavedenie role Product Owner Proxy zjednoduší prácu Product Ownerovi a všetky právomoci nebudú len na Product Ownerovi.

Samozrejme to bude závisieť na vzájomnej dohode, ale samotná komunikácia so zákazníkom, vytváranie cieľov projektu ostane na Product Ownerovi. Samotné prenesenie požiadaviek od zákazníka do formy tiketov v Backlogu môže byť delegované na človeka, ktorý bude na pozícii Product Owner Proxy. Výhodou je samozrejme aj to, že tento človek bude v kontakte s vývojovými tímami a bude pre nich ľahšie dostupný. Nakoľko tento človek nebude mať na starosti všetky kompetencie Product Ownera, tak bude jeden Product Owner Proxy pridelený dvom tímom. Samozrejme, že zodpovednosť Product Proxy Ownera je podobne veľká ako tomu je pri samotnom Product Ownerovi.

**Tabuľka 10: Návrh pozície Product Owner Proxy** (Zdroj: vlastné spracovanie)

	<b>Nový zamestnanci</b>	<b>Aktuálny zamestnanci firmy</b>
<b>Počet Product Owner Proxy</b>	1	3

V rámci firmy je výhodné, že budú túto pozíciu zastávať zamestnanci, ktorí napríklad pôsobili ako projektoví manažéri, tím lídri alebo architekti nakoľko ľudia na týchto pozíciách majú celkovo dobrú znalosť produktu, jeho funkcií a poznajú produkt aj po technickej stránke.

Podobne ako tomu bolo pri Product Ownerovi, aj v rámci tejto role ju budú zastávať traja interní zamestnanci a jeden nový. Títo zamestnanci absolvujú 3-dňové školenie spolu s ľuďmi na pozícii Product Owner. Obsah školení bude totožný a bude potrebné,

aby ho všetci absolvovali a pochopili princípy Scrumu, ktoré sa výrazne líšia od kompetencií, ktoré zastávali doteraz.

Product Owner Proxy bude mať vo firme iXperta zodpovednosť za:

- Formálnu tvorbu Backlogu – vytváranie úloh v Jira Software.
- Byť prítomný na meetingoch so Scrum Tímom.
- Komunikácia s Product Ownerom a zákazníkom – väčšinu času vzdialene pomocou komunikačného programu, takže bude nutnosťou komunikovať v angličtine.

Nakoľko Product Owner Proxy nebude mať v kompetencii všetky úlohy prislúchajúce Product Ownerovi, môže sa zapájať aj do riešení samotných úloh, ale nemalo by to byť jeho prioritou.

### **3.1.4 Scrum Team Member**

Oproti aktuálnemu zloženiu tímov v tradičných metódach sa v rámci Scrum tímov nebudú rozlišovať jednotliví členovia na architektov, analytikov, testerov a kodérov ale budú mať univerzálnu úlohu.

Základnými úlohami člena tímu je plnenie jednotlivých úloh vybraných zo Sprint Backlogu, kde si úlohy vyberá sám a je zodpovedný za výslednú podobu software. Jeho povinnosťou je dokončenie tejto úlohy do konca sprintu tak, aby boli splnené požiadavky zákazníka. Formu, štýl ako bude software vyvíjaný, a voľbu nástrojov si bude vývojár voliť sám, pri tom bude mať voľnú ruku, ale výsledok bude jasne daný požiadavkami zákazníka.

Cieľom návrhu pri tejto role je pripraviť každého člena tímu tak, aby mohol zastúpiť kolegu v prípade absencie alebo neprítomnosti, či už sa bude jednať o testera alebo človeka, ktorý sa viac orientuje na databázy. Ale so zavedením agilného riadenia sa mení prístup k celému procesu vývoja aj z pohľadu vývojárov. Každý člen vývojového tímu absolvuje v skupinkách po dvoch tímoch 2-denný kurz základov princípu Scrumu, kde by malo byť vysvetlené ako celý proces prebieha, od definície jednotlivých meetingov až po samotné artefakty Scrumu.

Súčasťou návrhu bude aj vzhľadom na produkt, ktorým sa spoločnosť zaoberá a vyvíja, snaha o vyvolanie väčšej interakcie samotných členov tímu. Môže to napomôcť vývoju

nových funkcií na aktuálne vyvíjanom produkte, a tieto pripomienky môžu byť pre firmu do budúcnosti veľmi cenné. Je to jedna z vecí, ktorá bola v analýzach spomínaná a kde by firma mohla nájsť nové príležitosti.

### **3.1.5 Zákazník**

Dôležitou a ústrednou súčasťou celého procesu vývoja by mal byť zákazník. V prípade iXperty je zákazníkom zahraničná firma, pre ktorú je software vyvíjaný. Interakcia zákazníka so Scrum tímami prostredníctvom Product Ownera a Product Owner Proxy bude kľúčová pre úspešnosť projektu.

Hlavnými a najdôležitejšími úlohami, ktoré bude zákazník zastávať:

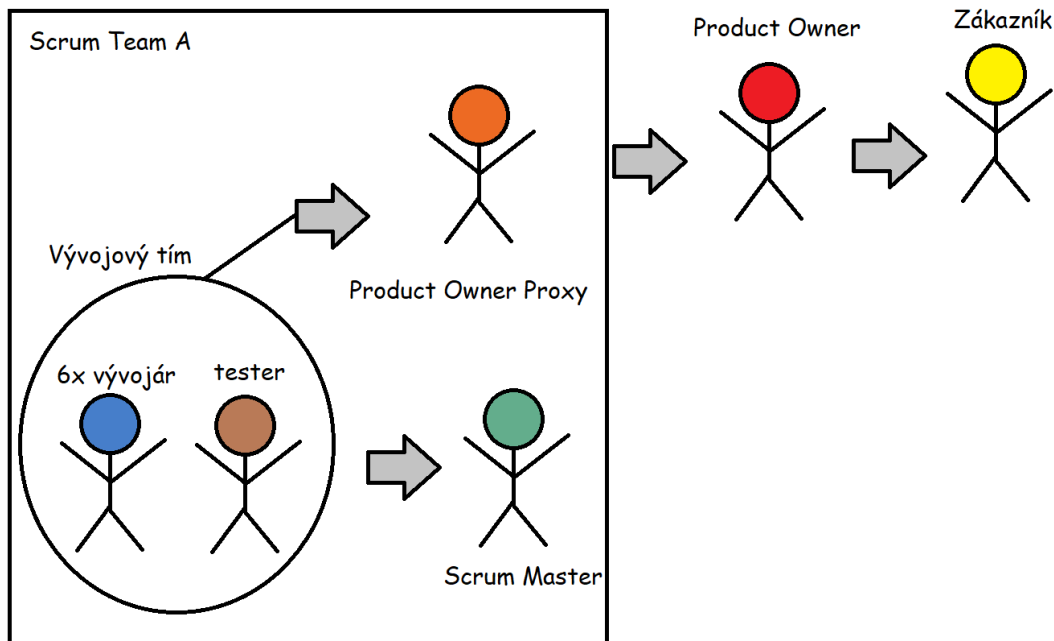
- Definovanie úloh – zákazník bude popisovať požiadavky svojej úlohy a čo by malo byť výsledkom zadanej úlohy (definovanie nových funkcií softwaru).
- Účasť na Sprint Review prostredníctvom videokonferencií.
- Testovanie aplikačných požiadaviek – zákazník poskytuje vývojárom spätnú väzbu na vyvíjaný software. V prípade pripomienok je nutné tieto pripomienky povedať Scrum tímu včas.

Bude potrebné, aby sa Product Owner stretával so zákazníkom na meetingoch, a dohodnuté požiadavky boli prenesené do User Stories. Budú stanovené pravidelné meetingy, tak aby sa nestratil kontakt so zákazníkom a bol stále v obraze. Organizovanie stretnutí so zákazníkom bude vhodné naplánovať dostatok času pred Groomingom, aspoň tak aby bol celý Sprint čas na formulovanie User Stories.

Je potrebné myslieť na to, aby sa vývojový tím alebo Product Owner nestal tým, kto bude definovať danú úlohu namiesto zákazníka. Pre úspešnosť vývoja projektu pomocou Scrumu je nutné, aby sa aktívne zúčastnili všetky strany. Podstatou Scrumu je predsa úspešnosť projektu, ktorá by mala byť prvoradá pre všetkých, ktorí sú zaangažovaný, a bez záujmu zákazníka to nepôjde.

### 3.1.6 Scrum Team

Do Scrum tímu budú patriť ľudia, ktorí budú zastávať role, ktorých úlohy boli presne zadefinované a určené doteraz. Vytvorený návrh fungujúceho Scrum tímu v spoločnosti by mal vyzeráť podobne ako na obrázku.



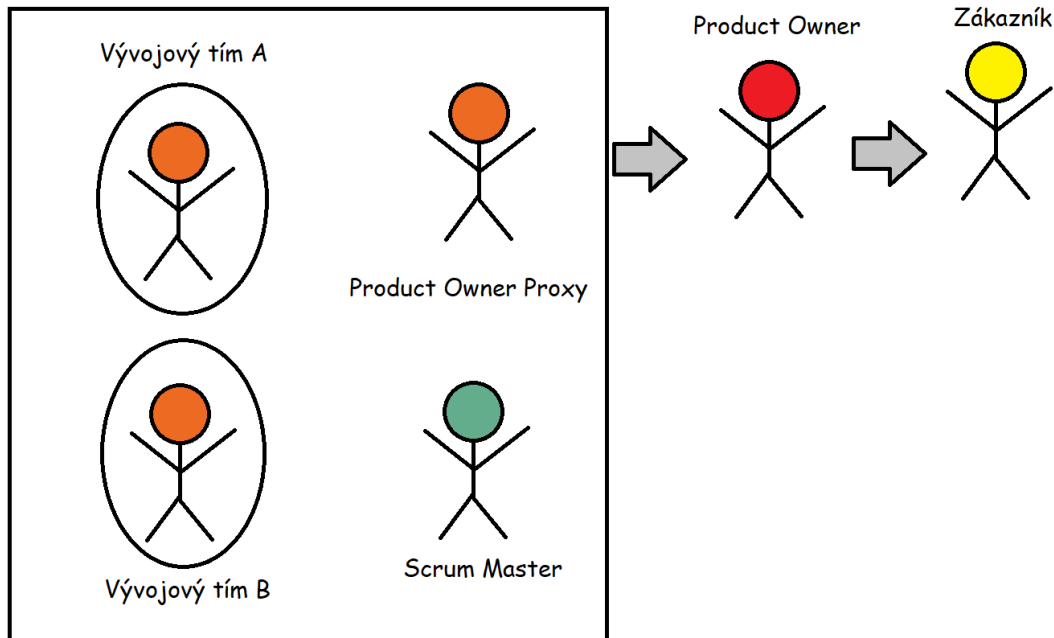
Obrázok 14: Návrh Scrum Tímu (Zdroj: vlastné spracovanie)

Scrum Team bude pozostávať z 8 až 10 členov, kde samotný vývojový tím bude tvoriť 6 až 8 vývojárov, ktorí budú mať výkonnú funkciu a Scrum Master a Product Owner Proxy, ktorí bude úzko spolupracovať s Product Ownerom ako takým. Product owner nebude v priamom kontakte s tímom, nakoľko sa fyzicky nachádza v inom štáte preto bolo potrebné zaviesť rolu Product Owner Proxy, ktorý sa bude podieľať na práci Product ownera a hlavne komunikovať v rámci Scrum Teamu.

Cieľom je vytvoriť tím, kde budú členovia vzájomne zastupiteľní, takže nútená absencia nejakého člena výrazne neovplyvní prácu tímu na jednotlivých úlohách. Je dôležité, aby celý tím pracoval na splnení Sprintu spoločne, tak aby bol dokončený úspešne. Do každého tímu bude pridelený tester z testerského tímu, ktorý bude vykonávať hlavne funkciu testera v rámci Scrum tímu. Je jasné, že jeho objem práce bude väčší ku koncu Sprintu. Preto sa vždy prvé dni Sprintu môže podieľať na analyzovaní úloh a ohraničovaní kritických scenárov pre vývojárov. Podobne ku koncu Sprintu sa vývojári



budú podieľať na testovaní dokončených úloh, tak aby sa Sprint stihol dokončiť, a presne toto je cieľom Scrumu. Tým pádom aj pri neočakávanej absencii testera v Scrum tíme si bude vedieť tím poradiť pri testovaní aj sám. Tento model zloženia tímu bude aplikovaný na postupnú transformáciu všetkých tímov vo firme.



**Obrázok 15: Product Owner Proxy a Scrum Master a dva vývojové tímy** (Zdroj: vlastné spracovanie)

Na obrázku je zobrazený návrh, kde bude Product Owner Proxy a Scrum Master spoločný pre dva vývojové tímy. Toto riešenie by malo byť menej nákladné ako keby mal každý tím svojho Scrum Mastra a Product Ownera. Tak isto aj v rámci plnenia svojich úloh by mali mať dosť času plniť ich pre oba tímy súčasne.

## 3.2 Návrh meetingov a základných artefaktov Scrumu

### 3.2.1 Backlog

Tak ako už bolo povedané, o Backlog sa bude starať človek s rolou Product Ownera. V jednotlivých tímoch firmy na tom budú spolupracovať najmä Product Owner s Product Owner Proxy, ktorého úlohou bude najmä vytváranie tiketov na základe požiadaviek, ktoré zistil priamo Product Owner od zákazníka. Dalo by sa povedať, že samotný Product Owner bude mať na starosti hlavne získanie hlavných myšlienok od zákazníka, smerovanie a definovanie cieľov pre produkt, ale samotnú formálnu tvorbu

Backlogu bude mať na starosti Product Owner Proxy. Táto forma bude pre fungovanie tímov vo firme efektívnejšia a jednoduchšia, ako keby bol na všetko Product Owner sám.

Backlog bude pozostávať z tiketov, ktoré budú tvorené najmä človekom s rolou Produkt Owner Proxy. Ale na vytváraní tiketov sa budú môcť podieľať aj jednotliví členovia vývojových tímov. Slovo tiket je použité zámerne a bude bližšie vysvetlené pri zavedení Jira Software. Tiketom je myslené prevedenie danej úlohy do elektronickej podoby v Jira Software, ktorý bude používaný. Samotný Backlog môže byť tvorený viacerými formami tiketov. Tu je princíp Scrumu trochu upravený v spoločnosti iXperta podľa toho, aký produkt vyvíja, a aký typ práce prevažne vykonáva. V náväznosti so spomínanou Jirou bude možné rozlišovať jednotlivé úlohy v Backlogu na:

- User Story
- Epic
- Task
- Bug

Podrobnejšie rozlišovanie týchto úloh bude vysvetlené v podkapitole so zavedením Jira Software. Časťou Backlogu by mali byť úlohy, ktoré budú prinášať zákazníkovi hodnotu. Najväčšiu hodnotu bude pre zákazníka predstavovať Epic a User Story.

Nakoľko sa pripravuje vývoj softwaru na nový model telefónu, bude potrebné vytvoriť Backlog s novými úlohami a ich popismi. Pre zovšeobecnenie bude nový model telefónu nazývaný CPxy a starý model CPx. Súčasťou tohto Backlogu budú zo začiatku definované nasledujúce úlohy (User Story):

#### **Návrh Produktového Backlogu:**

- CPxy: Boot CPxy telefónu
- CPxy: Prihlásenie CPxy telefónu v sieti
- CPxy: Urobiť základný hovor CPxy telefónom
- CPxy: Konferenčný hovor s viacerými užívateľmi
- CPxy: Zobrazenie nového užívateľského menu
- CPxy: Zobrazenie nového admin menu
- CPxy: Zobrazenie histórie konverzácií

- CPxy: Integrovanie Bluetooth komponenty
- CPxy: Spojenie CPx telefónu s ďalším zariadením pomocou Bluetooth
- CPxy: Prijímanie súborov cez Bluetooth
- CPxy: Spojenie CPx telefónu s bezdrôtovým Bluetooth headsetom
- CPxy: základný hovor CP telefónu s pripojeným Bluetooth headsetom s ďalším zariadením
- CPxy: konferenčný hovor CP telefónu s pripojeným Bluetooth headsetom s ďalšími zariadeniami

Vytvorenie takéhoto Backlogu je súčasťou tzv. Backlog Refinement Meeting, ktorý bude spomenutý v podkapitole nižšie.

### **Epic a User Story**

Epic bude predstavovať veľký funkčný celok, ktorý bude pozostávať z jednotlivých User Stories. User Stories budú menšie dielčie celky, ktoré budú dokopy vytvárať cieľ pôvodného Epicu. Je potrebné myslieť na to, že pri User story sa v spoločnosti bude jednať hlavne o novej funkcionalite pre existujúci software, a pod Epic sa dá predstaviť napríklad vývoj nového modelu telefónu. Definovanie User Stories a Epics je v plnej réžii Product Owner alebo Product Owner Proxy na základe užívateľských požiadaviek. Príkladom User Story, ktorá bude využitá pri vyvíjaní nového modelu telefónu bude napríklad samotný štart telefónu:

(z Backlogu - User Story CPxy: Boot CPxy telefónu)

**AKO užívateľ CPxy telefónu CHCEM zapnúť telefón tak, aby sa**  
*zobrazila úvodná obrazovka*

(z Backlogu - User Story CPxy: Spojenie CPx telefónu s bezdrôtovým Bluetooth headsetom)

**AKO užívateľ CPxy telefónu CHCEM pripojiť bezdrôtový headset cez**  
*Bluetooth tak, aby som ho videl v menu pripojený*

Dôležitou časťou definovania role budú aj akceptačné kritéria, za akých sa bude dať považovať daná User Story za dokončenú. Akceptačné kritéria sú stanovené na základe Product ownera, ktorý ich stanoví podľa vlastného uváženia z požiadaviek zákazníka.

V našich dvoch uvedených príkladoch by boli akceptačné kritéria pri prvej User Story napríklad to, že telefón sa nevypne a nenastane žiadna chyba. Pri druhej User Story bude akceptačným kritériom napríklad to, že bude jasné z užívateľského menu, že headset je pripojený, a aj to, že samotné zariadenie sa bude správať tak že pripojené.

### **Task**

Bude sa jednať najmä o jednorazovú činnosť, ktorá bude hovoriť o tom, aké činnosti je potrebné urobiť na to, aby sme boli schopný zákazníkovi doručiť hotovú User Story.

V našom prípade sa bude jednať napríklad o User Story „CPxy: Boot CPxy telefónu“ z vyššie spomenutého Produktového Backlogu, ktorá bude rozdelená na menšie Tasky.

### **User Story CPxy: Boot CPxy telefónu**

User Story bude rozdelená do nasledujúcich menších Taskov:

- Upravenie makefile súboru.
- Preloženie zdrojového kódu pre nový model telefónu.
- Upravenie skriptu na spustenie telefónu s novým softwarom.
- Nahratie nového software do nového modelu telefónu.

### **Bug**

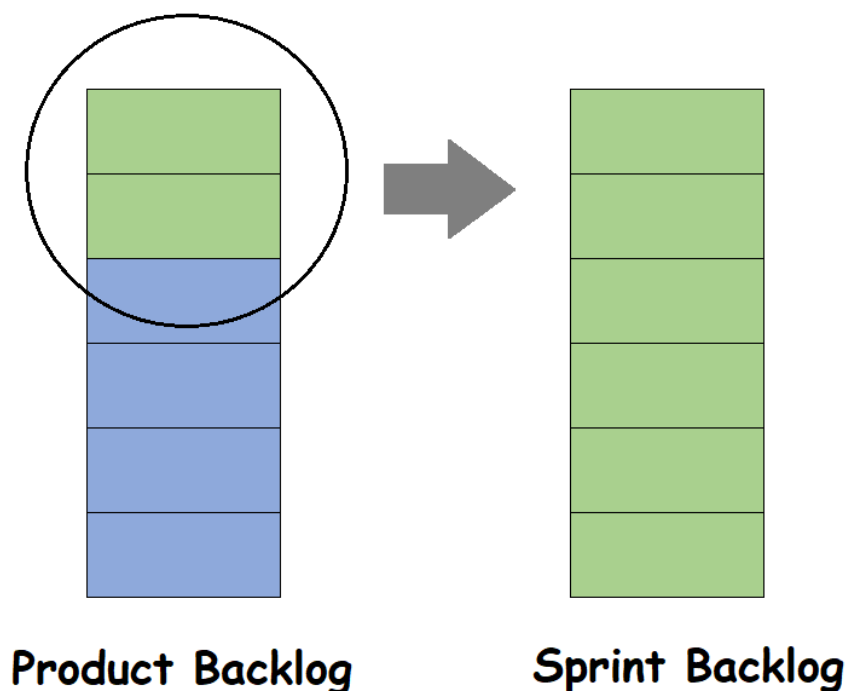
V tomto prípade sa jedná úlohu, ktorá môže, byť vytvorená nejakým členom tímu (najčastejšie testerom), ale aj zákazníkom, ktorý našiel chybu v existujúcej verzii software. Buggy budú zahrnuté do Produktového Backlogu v rámci Backlog Refinement Meeting. Príkladom chyby nájdenej testermi na predchádzajúcom modeli telefónu je napríklad nasledujúci problém:

- CPx: Nefunguje podržanie (Hold) hovoru na CPx telefóne

Buggy vyžadujú analýzu kódu a nájdenie daného problému s následným opravením. Na rozdiel od User Story im nebudú pridelené Story pointy.

V tímoch sa budú používať dva Backlogy tak, ako je to pôvodne v Scrum zadané:

- Product Backlog – zoznam úloh, ktoré je nutné v rámci vývoja implementovať.
- Sprint Backlog – zoznam úloh, ku ktorému sa tím zaväzuje, že budú koncom Sprintu vyriešené.



**Obrázok 16: Product Backlog a Sprint Backlog** (Zdroj: vlastné spracovanie podľa [11])

Podstatný rozdiel medzi nimi je ten, že do Sprint Backlogu nebude možné po zahájení Sprintu pridávať ani odoberať úlohy. Do Product Backlogu sa veci môžu, a aj by sa mali pridávať.

Nakoľko je v spoločnosti veľa tímov ktoré vyvíjajú ten istý produkt, akurát sa podieľajú na rozdielnej časti software, je takmer isté, že budú využívať jeden Product Backlog. Každý tím v spoločnosti bude riešiť úlohy z vlastnej oblasti, takže bude dochádzať k minimálnemu prekryvaniu úloh v Backlogu viacerými tímami. Jednoducho sa to dá predstaviť tak, že v rámci jedného Backlogu si budú tímy brať do Sprintov úlohy z oblastí, ktorým sa najviac venujú a venovali doteraz.

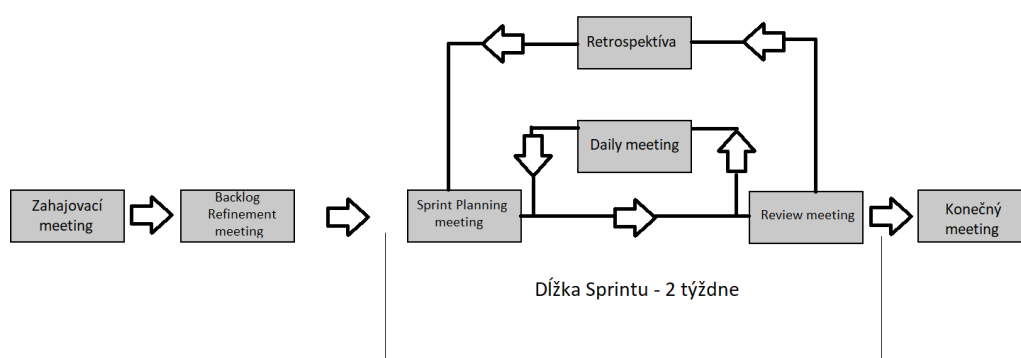
### 3.2.2 Sprint

Neoddeliteľnou súčasťou Scrumu sú Sprints. Nastavenie dĺžky sprintov je z pohľadu celého procesu kľúčové a zásadné. Väčšinou sa nastavenie dĺžky Sprintu udáva na základe toho, ako často chceme zákazníčkovi ukazovať riešenia a byť s ním v kontakte a ako často sa menia požiadavky na vyvíjaný produkt. Sprint, ktorý začne, by nemal byť

rušený ničím iba koncom. V núdzovom stave môže Sprint zrušiť Product Owner, ale musí k tomu mať veľmi vážny dôvod.

Je potrebné myslieť na to, že výsledkom každého Sprintu by mala byť dokončená verzia software, ktorá bude spustiteľná a validovateľná zákazníkom. Na základe jeho spätnej väzby zistíme, či daná verzia software spĺňa kritéria zákazníka, alebo je potrebné zmeniť požiadavky do ďalšieho Sprintu.

Dĺžka sprintu by sa nastavila pre všetky tímy na 2 týždne. Táto dĺžka by mala byť postačujúca na prípadné odhalenia problémov v implementáciách a včasnú reakciu zákazníka na prípadné odlišnosti v správaní testovanej verzie. Samozrejme, v priebehu dvoch týždňov bude potrebné absolvovať meetingy, ktoré sú v Scrum potrebné, ale aj tie by sa dali považovať za plusy, pretože v prípade zmeny požiadaviek budú tímy schopné reagovať rýchlejšie. Je jasné, že vývojári si na túto zmenu budú musieť zvyknúť ale je potrebné pochopiť, že je to za účelom dosiahnutia úspechu každej strany, ktorá je do projektu zahrnutá.



**Obrázok 17: Konkrétne nastavenie dĺžky Sprintu** (Zdroj: vlastné spracovanie podľa [12])

Neodmysliteľnou súčasťou Scrumu sú meetingy, bez ktorých by Scrum určite nefungoval. V nasledujúcej kapitole budú predstavené jednotlivé návrhy na priebeh, obsah a dôvod meetingov.

### 3.2.3 Zahajovací meeting

Z pohľadu existujúceho software je zahajovací meeting v spoločnosti dôležitý hlavne z pohľadu zoznámenia Scrum tímov so zmenami štýlu riadenia a zavedenia Scrumu. Členovia vývojových tímov majú väčšinu vecí zažitých z predchádzajúceho procesu a

produkt viac-menej poznajú. Preto je zahajovací meeting vhodný v tomto prípade hlavne na predstavenie Product Ownera, ktorý predstaví svoju víziu o projekte podľa toho ako sa dohodol so zákazníkom. Podobne tento meeting slúži na vzájomné zoznámenie jednotlivých členov Scrum tímov a predstavenie kompetencií Scrum Mastrom. Pre prípadných nových členov vývojových tímov môže byť tento meeting dôležitý z pohľadu poznania produktu, na ktorom budú pracovať.

Súčasťou tohto meetingu môže byť v našom prípade aj predstavenie predbežného Backlogu, ktorý je tvorený z vízie Product Ownera pri vyvíjaní novšieho modelu telefónu, kde by sa mali nachádzať nejaké novinky. V tomto prípade je predstavenie predbežného Backlogu členom vývojového tímu zásadné, pretože oni sa vedia na túto víziu pozrieť z technického hľadiska a vedia odhadnúť, či je tím schopný dosiahnuť požadované ciele.

Naopak ak spoločnosť plánuje do budúcnosti získať nové projekty a pracovať na nich, tak tento meeting bude pomerne dôležitý na bližšie predstavenie produktu a produktového backlogu Product Ownerom.

### **3.2.4 Backlog Refinement Meeting**

Tento meeting je dôležitý vo firme hlavne z pohľadu transformácie požiadaviek a úloh jednotlivých tímov do Produktových Backlogov. Na dôležitosti mu pridáva aj to, že firma bude vyvíjať software na nový model telefónu. K tomu bude treba zdefinovať vízie Product Ownera na základe požiadaviek zákazníka a preniesť ich do Produktového Backlogu. Produktový Backlog by mal v priebehu Refinement Meetingu obsahovať:

- Transformované úlohy (bugy), zo starého systému, ktorý spoločnosť používala – je nevyhnutné tieto úlohy preniesť do produktového backlogu, nakoľko sa jedná o chyby, ktoré sa nachádzajú v aktuálnom software
- Pridať nové User Stories, ktoré sa týkajú nového modelu telefónu

Je potrebné spomenúť, že tento meeting budú využívať členovia vývojového tímu na prípadné otázky Product Ownerovi, ktoré v špecifikácii chýbajú alebo nie sú jasné. Product Owner by mal byť schopný na tieto otázky odpovedať. Pokiaľ na otázky odpovedať nevie, tak by sa na danej User Story nezačne pracovať. Na tomto meetingu bude potrebné, aby bol Product Owner v priamom kontakte s vývojovým tímom a

nielen s Product Owner Proxy. Takže buď na tento meeting Product Owner pricestuje alebo sa bude konať cez internet v podobe videokonferencie.

Po tomto meetingu by mal Product Owner spolu s Product Owner Proxy byť schopný určiť priority jednotlivým úlohám v hotovom Produktovom Backlogu. Priority budú pre jednotlivé úlohy v tímoch určené od P0, P1, P2, P3, P4.

Najväčšiu prioritu budú mať úlohy s označením P0, a tá bude značiť, že daná úloha je kritická a je nutné ju zobrať do Sprintu.

V našom prípade bude Produktový Backlog vytvorený z User Story, ktoré boli spomenuté v predošlej podkapitole (4.2.1 Backlog) a pridaných úloh prevažne chýb, ktoré boli objavené v staršej verzii softwaru, Nový model telefónu budeme nazývať CPxy a starý CPx, aby sme ich vedeli rozlíšiť v uvedenom príklade Backlogu. Priority určené Product Ownerom a Product Ownerom Proxy sú zvýraznené a zobrazené v zátvorke za danou úlohou.

**Tabuľka 11: Produktový Backlog s určenými prioritami** (Zdroj: vlastné spracovanie)

<b>Produktový Backlog</b>	
<b>Užívateľské príbehy (User Stories)</b>	<b>Chyby (Bugs)</b>
<ul style="list-style-type: none"> <li>• Cpxy: Boot CPxy telefónu - <b>(P0)</b></li> <li>• Cpxy: Prihlásenie CPxy telefónu v sieti - <b>(P1)</b></li> <li>• CPxy: Urobiť základný hovor CPxy telefónom - <b>(P2)</b></li> <li>• CPxy: Konferenčný hovor s viacerými užívateľmi - <b>(P2)</b></li> <li>• CPxy: Zobrazenie nového užívateľského menu - <b>(P2)</b></li> <li>• CPxy: Zobrazenie nového admin menu - <b>(P2)</b></li> <li>• CPxy: Zobrazenie histórie konverzácií - <b>(P3)</b></li> </ul>	<ul style="list-style-type: none"> <li>• CPx: Konferenčný hovor (5 účastníkov) bez audia – <b>(P0)</b></li> <li>• CPx: Konferenčný hovor s pripojeným headsetom bez audia v headsete - <b>(P0)</b></li> <li>• CPx: Nefunguje podržanie (Hold) hovoru CPx telefónu - <b>(P1)</b></li> <li>• CPx: Zle zobrazené meno účastníka v konferencii - <b>(P0)</b></li> <li>• CPx: Nie je možné napárovať iPhone - <b>(P1)</b></li> <li>• CPx: Zle zobrazené položky užívateľského menu - <b>(P1)</b></li> </ul>



<ul style="list-style-type: none"> <li>• CPxy: Integrovanie Bluetooth komponenty- <b>(P3)</b></li> <li>• CPxy: Spojenie CPx telefónu s ďalším zariadením pomocou Bluetooth</li> <li>• CPxy: Prijímanie súborov cez Bluetooth - <b>(P3)</b></li> <li>• CPxy: Spojenie CPx telefónu s bezdrôtovým Bluetooth headsetom - <b>(P3)</b></li> <li>• CPxy: základný hovor CP telefónu s pripojeným Bluetooth headsetom s ďalším zariadením - <b>(P3)</b></li> <li>• CPxy: konferenčný hovor CP telefónu s pripojeným Bluetooth headsetom s ďalšími zariadeniami - <b>(P3)</b></li> </ul>	<ul style="list-style-type: none"> <li>• CPx: Nie je možný prenos .png obrázkov cez Bluetooth – <b>(P2)</b></li> <li>• CPx: Nie je možný prenos .wave audio súborov cez Bluetooth - <b>(P2)</b></li> <li>• CPx: Obrázok pri kontakte sa nezobrazuje správne - <b>(P3)</b></li> </ul>
--	--

### 3.2.5 Grooming

Z pohľadu Scrum tímov vo firme bude tento meeting veľmi nápomocný a kľúčový pre plánovanie. Nakoľko architektúra aktuálneho software je pomerne zložitá, tak tieto meetingy by mali slúžiť na ujasnenie čo jednotlivé User Stories budú robiť jak náročné ich bude implementovať. Podstatou tohto meetingu bude ohodnocovanie jednotlivých User Stories členmi tímu. Existujú rôzne spôsoby ohodnocovania User Stories. Jednou z nich je využitím Planning Poker kariet, ktoré by mali byť vhodným variantom. Každá karta bude určovať istý počet bodov, ktoré sa budú nazývať Story Pointy. Balíček bude obsahovať karty s hodnotami 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100, nekonečno, ?, a coffee break karta.



Obrázok 18: Planning Poker karty (Zdroj: [12])

Na tomto meetingu je potrebné, aby členovia vývojového tímu vyjadrili svoj názor a ohodnotili jednotlivé User Stories na základe svojich skúseností a poznania produktu. Scrum Master bude vystupovať ako moderátor a bude viesť meeting tak, ako pri väčšine meetingov. Prítomnosť samotného Product Ownera nemusí byť nutnosťou, ale v prípade nejasností bude vhodná. Ale človek na pozícii Product owner Proxy by mal byť určite na meetingu prítomný.

Na ohodnocovanie Produktového Backlogu by sa použili Planning Poker karty. V prípade nášho produktového Backlogu by jednotlivé User Story boli ohodnotené nasledovne.

#### Ohodnotenie Produktového Backlogu s prioritami:

- Cpxy: Boot CPxy telefónu - **(P0) – 20 Story Points**
- Cpxy: Prihlásenie CPxy telefónu v sieti - **(P1) - 13 Story Points**
- CPxy: Urobiť základný hovor CPxy telefónom - **(P2) – 13 Story Points**
- CPxy: Konferenčný hovor s viacerými užívateľmi - **(P2) – 8 Story Points**
- CPxy: Zobrazenie nového užívateľského menu - **(P2) – 8 Story Points**
- CPxy: Zobrazenie nového admin menu - **(P2) – 8 Story Points**

- CPxy: Zobrazenie histórie konverzácií - **(P3) - 5 Story Points**
- CPxy: Integrovanie Bluetooth komponenty- **(P3) - 20 Story Points**
- CPxy: Spojenie CPx telefónu s ďalším zariadením pomocou Bluetooth - **20 Story Points**
- CPxy: Prijímanie súborov cez Bluetooth - **(P3) - 13 Story Points**
- CPxy: Spojenie CPx telefónu s bezdrôtovým Bluetooth headsetom - **(P3) - 20 Story Points**
- CPxy: základný hovor CP telefónu s pripojeným Bluetooth headsetom s ďalším zariadením - **(P3) - 13 Story Points**
- CPxy: konferenčný hovor CP telefónu s pripojeným Bluetooth headsetom - **(P3) - 8 Story Points**

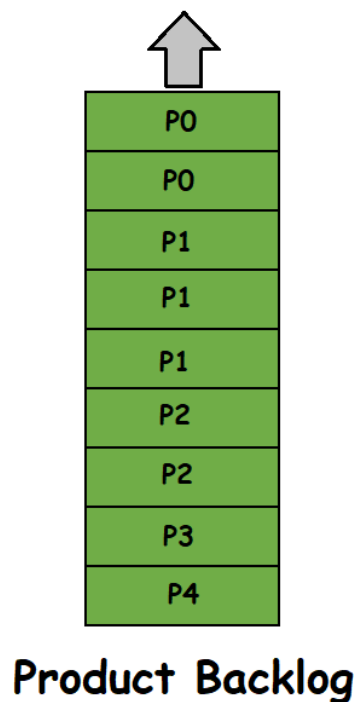
Časové naplánovanie tohto meetingu závisí na samotnom Scrum tíme, no odporúča sa jeho naplánovanie v prvej časti Sprintu. Dôvodom je prípadná zmena User Stories v produktovom Backlogu, ktorá by mohla ovplyvniť plánovanie do ďalšieho Sprintu.

Veľmi dôležitou vecou je ohodnocovať úlohy na základe relatívneho odhadu medzi jednotlivými User Stories navzájom. V prípade, že nejaká User Story bude mať príliš veľké hodnotenie, je potrebné sa zamyslieť nad jej rozdelením na menšie časti. V žiadnom prípade by sa nemalo stať, že úlohy budú ohodnocované na základe časovej náročnosti, to je proti princípom Scrumu a je potrebné, aby si na to dávali jednotlivé tímy obrovský pozor. Výsledkom Groomingu by malo byť bodové ohodnotenie každej User Story tak, ako je to uvedené na príklade nášho Produktového Backlogu. Naopak chyby (Bugs) nájdené v jednotlivých verziách software sa ohodnocovať nebudú, preto nebudú zahrnuté ani na Groomingu.

### 3.2.6 Plánovanie

Plánovania by sa mali zúčastniť tímy, ktoré absolvovali Grooming, a vedia, o čo v jednotlivých User Stories ide. Je potrebné si dať pozor na to, aby plánovanie začalo tým, že Product Owner s Product Owner Proxy určia priority jednotlivým úlohám a zoradia ich v Backlogu od tých najprioritnejších zhora-dole. Product Owner a Product Owner Proxy by mali nastaviť priority podľa toho, aby boli splnené kritické zákaznícke požiadavky prioritne. Typicky sa v konkrétnom prípade tímov spoločnosti bude jednať o nové funkcie telefónov, ktoré by zákazník chcel používať.

Nakoľko má tím jednotlivé User Story ohodnotené už z Groomingu, jednotlivé celky by už mali byť jednoznačné a pochopiteľné na implementáciu. Napriek tomu by sa mal tím posledný krát zamyslieť, či náhodou nejaká User Story neobsahuje nejasnosti. Tím si bude vyberať jednotlivé User Stories podľa priorít, ako ich určil Product Owner s Product Owner Proxy. Hlavným rozdielom oproti normálnemu Sprintu v Scrum je to, že sa budú plánovať aj chyby, ktoré sa našli vo verzii software (tzv. buggy). Jedná sa o prispôsobenie princípov metodiky Scrum tímom spoločnosti iXperta, nakoľko fixovanie chýb patrí tiež k podstatnej časti vývojových tímov.



**Obrázok 19: Priority v Produktovom Backlogu** (Zdroj: vlastné spracovanie)

Určite je potrebné myslieť na to, že na začiatku bude ťažké odhadnúť koľko jednotlivé tímy za 2-týždňový Sprint stihnú. Postupom času by každý tím mal byť schopný odhadnúť koľko bodov User Stories je schopný stihnúť. Velocitu tímu by mal sledovať Scrum Master a on by mal byť tým človekom, ktorý by mal usmerňovať tím k tomu, aby bola táto hranica bodov User Stories čo najviac ustálená a Sprint nebol preplánovaný. Preto bude úloha Scrum Mastra dôležitá v priebehu plánovania.

Z časového hľadiska je celkovo na tíme, kedy si tento meeting naplánuje. Je ale potrebné, aby sa plánovania zúčastnili všetci členovia tímu a predchádzal mu Grooming. Takže konkrétnym a dobrým návrhom na naplánovanie tohto meetingu by mohla byť napríklad streda.

### Naplanovanie Sprintu 1

Na základe Backlog Refinement meetingu a Groomingu, kde bol zostavený Produktový Backlog s ohodnotením jednotlivých User Stories bude zobrazený plán prvého Sprintu, do ktorého budú zahrnuté User Story s vysokou prioritou, ale taktiež bugy, ktoré je potrebné vyriešiť a sú prioritné.

**Tabuľka 12: Naplanovanie Sprintu 1** (Zdroj: vlastné spracovanie )

<b>Sprint 1 Backlog</b>	
<b>Užívateľské príbehy (User Stories)</b>	<b>Chyby (Bugs)</b>
<ul style="list-style-type: none"> <li>• Cpxy: Boot CPxy telefónu - <b>(P0) – 20 Story Points</b></li> <li>• Cpxy: Prihlásenie CPxy telefónu v sieti - <b>(P1) - 13 Story Points</b></li> <li>• CPxy: Urobiť základný hovor CPxy telefónom - <b>(P2) - 13 Story Points</b></li> </ul>	<ul style="list-style-type: none"> <li>• CPx: Konferenčný hovor (5 účastníkov) bez audia – <b>(P0)</b></li> <li>• CPx: Konferenčný hovor s pripojeným headsetom bez audia v headsete - <b>(P0)</b></li> <li>• CPx: Nefunguje podržanie (Hold) hovoru CPx telefónu - <b>(P1)</b></li> <li>• CPx: Zle zobrazené meno účastníka v konferencii - <b>(P0)</b></li> <li>• CPx: Nie je možné napárovať iPhone 7 - <b>(P1)</b></li> </ul>

Do prvého Sprintu budú naplánované 3 User Story (dokopy 56 Story Pointov) a päť chýb v starom modeli telefónu, ktoré bude potrebné vyriešiť. Na základe výsledkov dokončeného prvého Sprintu sa bude lepšie plánovať Sprint 2, pretože bude existovať odhad, koľko toho vie tím za Sprint splniť.

### 3.2.7 Scrum Meeting

Tak ako tomu bolo pri plánovaní a groomingu je pomerne dôležité, aby boli všetci členovia tímu prítomní na tomto meetingu. Nastavenie času jednotlivých Scrum meetingov bude nastavené na 9:30 ráno. Neskôr je možné tento čas zmeniť, ak by to nejakému tímu nevyhovovalo, ale je potrebné stanovený čas dodržiavať.

Je dobré rozdeliť jednotlivé User Story na menšie Tasky tak, ako napríklad pri User Story „CPxy: Boot CPxy telefónu“ z Produktového backlogu:

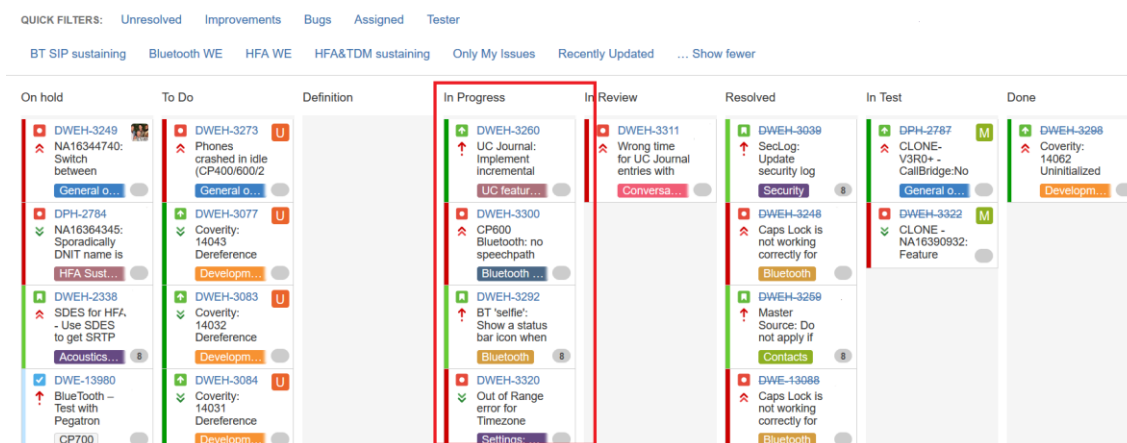
#### Tasky User Story - CPxy: Boot CPxy telefónu

- Upravenie makefile súboru – **dokončený**.
- Preloženie zdrojového kódu pre nový model telefónu – **pracuje sa na ňom**.
- Upravenie skriptu na spustenie telefónu s novým softwarom – **urobiť**.
- Nahratie nového softwaru do nového modelu telefónu – **urobiť**.

Rozdelenie User Stories na menšie časti tak, ako je to zobrazené na príklade, je vhodnejšie aj z pohľadu Scrum Meetingu, kde sú jednotlivé Tasky ľahšie uchopiteľné. Zo zobrazeného príkladu vidíme, ktoré Tasky sú dokončené, a na ktorých sa pracuje alebo by sa malo začať. Scrum Meeting bude začínať Scrum Master zhodnotením aktuálnej situácie v Sprinte. Potom slovo prevezmú jednotliví členovia, ktorí by mali zodpovedať na základné tri otázky, známe z teórie:

- Čo dokončil včera – aký Task dokončil včera.
- Na čom bude pracovať dnes – na akom Tasku bude pracovať dnes.
- Problémy, s ktorými sa stretol, a bránia mu Task dokončiť.

Scrum Master bude mať za úlohu moderovať tento meeting tak, aby nebol príliš dlhý, aby sa tam nerozoberali detaily, a hlavne či nenastali komplikácie pri plnení nejakej User Story, na ktorej pracuje člen tímu. Práve z hľadiska problémov, bude tento meeting dôležitý, pretože tím vie zhodnotiť, či sa aj napriek tomuto problému podarí ukončiť Sprint.



Obrázok 20: Návrh Scrum tabule (Zdroj: vlastné spracovanie)

Z pohľadu tohto meetingu nás bude zaujímať stav úloh v stĺpci In progress, ktorý Jira Software prehľadne zobrazuje a na obrázku je zvýraznený červenou farbou. Tieto úlohy by mali byť predmetom Scrum Meetingu.

Prípadné riešenie týchto problémov by nemalo prebiehať cez Scrum Meeting, ale až po ňom, najlepšie hneď po meetingu. Tím na ňom môže zhodnotiť, či úloha, kde sa vyskytol problém, vyžaduje viac členov na jej riešenie, prípadne nejako inak zakročiť, tak aby sa Sprint stihol.

V prípade spoločnosti iXperta sa bude hovoriť o účasti Product Owner Proxy a nie človeka v role Product Owner na Scrum Meetingu, pretože ten nie je fyzicky v kontakte s tímom. Účasť Product Owner Proxy na meetingu bude výhodou, pretože v prípade otázok sa ho členovia tímu môžu pýtať, ale nemal by zasahovať do riešenia jednotlivých úloh členov tímu a samotného meetingu.

### 3.2.8 Retrospektíva

Z pohľadu tímov vo firme, je dôležité povedať, že veľakrát dochádza k tomu, že v rámci niektorých úloh jednotlivé tímy spolupracujú s tímami aj v iných krajinách. Aby sa nezachádzalo do veľkých technických detailov, dôvodom je rozsiahlosť software, čo v skratke znamená, že každý tím má na starosti svoju časť (komponent). To má vplyv aj na priebeh tohto meetingu. Bude sa rozlišovať Retrospektíva globálna a Retrospektíva v rámci jednotlivých tímov.

#### Retrospektíva v tíme

Je potrebné správne navrhnuť priebeh Retrospektívy tak, aby to bola pre tím prospešná. Cieľom retrospektívy bude vyjadrenie názoru a pocitov každého člena tímu na dokončený Sprint. Retrospektíva má iba pár pravidiel, ktorých by sa mal tím držať ale spôsob prevedenia je už na každom tíme. Retrospektíva by sa mala držať týchto bodov a otázok, ktorých by sa mal každý člen držať:

- Čo bolo dobré, a čoho by sa mal tím držať?
- Čo bolo zlé, a s čím by mal tím prestať?
- Čo by bolo dobré zaviesť?

Samotné prevedenie retrospektívy môže prebiehať viacerými spôsobmi, ktorých existuje viacero. Každý tím ich môže vyskúšať viac, prípadne ich meniť, ak ich nejaký spôsob omrzí.

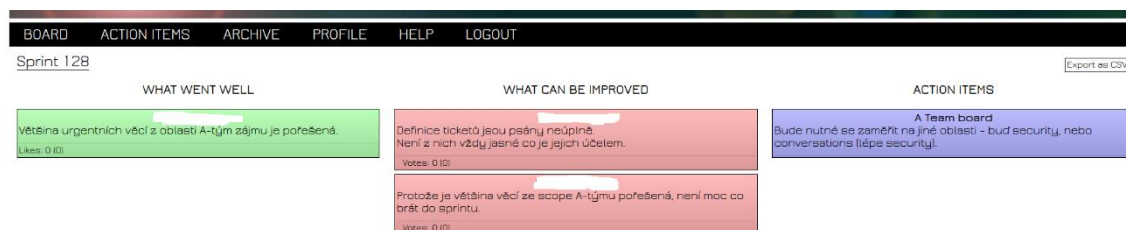
V rámci návrhu budú prezentované dva spôsoby, ktoré môžu tímy aplikovať. Prvým spôsobom je využívanie tabule s papierikmi, kde sa tabuľa rozdelí na štyri časti. Každý člen tímu dostane niekoľko papierikov s fixkami. Papieriky postupne jednotlivý členovia tímu lepia do jednotlivých častí tabule tak, ako sú zobrazené na obrázku, podľa toho kam si myslia, že patrí. Pomenovania papierikov by malo byť čo najstručnejšie a najvecnejšie. Papieriky s rovnakými myšlienkami sa lepia pod seba. Keď sa vystriedajú všetci členovia tímu, Scrum Master sa zaoberá najskôr tými bodmi, kde je nalepených pod sebou viacero papierikov. Tabuľka rozdelená na štyri kvadranty môže byť na tabuli prítomná počas celého Sprintu a papieriky sa na ňu môžu lepiť aj počas Sprintu, aby sa na niektoré veci nezabudlo.



Čo bolo dobré?	Čo bolo zlé?
Čo ponechať?	Čo vylepšiť?

**Obrázok 21: Rozdelenie tabule na retrospektívu** (Zdroj: vlastné spracovanie)

Druhým, viac technickým návrhom spôsobu Retrospektívy, je napríklad jednoduchý nástroj zobrazený na obrázku nižšie. Prvé dva stĺpce zobrazujú v podstate čo bolo v Sprinte dobré a čo zlé. V prípade, že nejaký člen súhlasí s už napísaným príspevkom, je možné súhlasiť a dať „Páči sa mi to“.



**Obrázok 22: Návrh aplikácie na retrospektívu** (Zdroj: vlastné spracovanie)

Tretí stĺpec na obrázku zobrazuje akciu, ktorú je potrebné vykonať na základe pripomienok z retrospektívy. Táto akcia (na obrázku Action Item) je výsledkom premýšľania celého tímu nad bodmi, ktoré majú najviac „Páči sa mi“ alebo papierikov nalepených pod sebou na tabuli. Túto debatu by mal moderovať Scrum Master a

smerovať členov tímu k tomu, aby sa venovali hlavne týmto bodom. Je dôležité, aby boli výstupom retrospektívy jednotlivé akcie, ktorých výsledkom by mali byť vylepšenia. Začiatkom ďalšej retrospektívy by mali byť pripomenuté veci, ktoré boli výsledkom predošlej retrospektívy, a hlavne, či boli vyriešené jednotlivé akcie. Je nutné, aby boli členovia pri retrospektíve aktívni, a v prípade, že to tak nie je, je nutné hľadať príčinu. Retrospektíva by mala byť naplánovaná na samotnom konci Sprintu, keď sú jednotliví členovia tímu schopní zhodnotiť celý Sprint.

### **Globálna Retrospektíva**

Tak ako už bolo spomenuté, vzhľadom na pomerne rozsiahlu štruktúru firmy je možné, že niektoré Akcie z retrospektív jednotlivých tímov budú siahať do iných tímov alebo na veci, ktoré sú spoločné (prípadné nezhody alebo problémy, ktoré je nutné riešiť). Podľa toho ako sa členovia jednotlivých tímov dohodnú so svojimi Scrum Mastrami, môžu niektoré akcie preniesť do globálnej retrospektívy.

Táto retrospektíva je návrhom konkrétne pre túto spoločnosť, vzhľadom na pomerne veľký počet tímov. Na tomto meetingu sa stretnú Scrum Mastri z jednotlivých tímov a prezentujú výstupy retrospektív zo svojich tímov a snažia sa z toho vyvodiť nejaké riešenia. Tento meeting môže byť naplánovaný na začiatku ďalšieho Sprintu. Je potrebné ešte dodať, že tento typ meetingu bude potrebné riešiť vzdialene pomocou videokonferencie vzhľadom na to, že niektoré tímy sa nachádzajú v iných krajinách.

### **3.2.9 Sprint Review**

Sprint Review je meeting, kde Scrum Tím ukazuje zákazníkovi funkčnosť jednotlivých úloh, ku ktorým sa zaviazal na začiatku Sprintu. Bude dôležité, aby tím prezentoval User Stories, ktoré sú kompletne uzavreté a otestované. O samotné prezentovanie funkcionality sa budú starať jednotliví členovia vývojového tímu alebo Scrum Master, a v prípade otázok od zákazníka môžu reagovať.

Výsledkom Sprint review by mala byť odozva Product Ownera a hlavne zákazníka, či implementované úlohy spĺňajú požiadavky zákazníka definované pred začiatkom Sprintu. V prípade, že zákazník požaduje zmeny v existujúcej funkčnosti, tak sú definované nové úlohy do Product Backlogu.

Tento meeting by mal prebiehať posledný deň Sprintu, aby mal Scrum Tím dostatok času na uzatvorenie User Stories a prípravu Sprint Review. Na Sprint Review by mal byť prítomný celý tím aj s Product Ownerom a zákazníkom. V dôsledku toho, že zákazník a Product Owner sú v rozdielnych krajinách, bude potrebné Sprint Review robiť pomocou videokonferencie. Sprint Review pre samotný Scrum Tím bude znamenať uzatvorenie aktuálneho Sprintu a predstavenie úloh nasledujúceho Sprintu.

**Tabuľka 13: Sprint 1 Review** (Zdroj: vlastné spracovanie)

<b>Sprint 1 Review</b>	
<b>To Do</b>	<b>Done</b>
	<ul style="list-style-type: none"> <li>• Cpxy: Boot CPxy telefónu - <b>(P0) – 20 Story Points</b></li> <li>• Cpxy: Prihlásenie CPxy telefónu v sieti - <b>(P1) - 13 Story Points</b></li> <li>• CPxy: Urobiť základný hovor CPxy telefónom - <b>(P2) - 13 Story Points</b></li> <li>•</li> <li>• CPx: Konferenčný hovor (5 účastníkov) bez audia – <b>(P0)</b></li> <li>• CPx: Konferenčný hovor s pripojeným headsetom bez audia v headsete - <b>(P0)</b></li> <li>• CPx: Nefunguje podržanie (Hold) hovoru CPx telefónu - <b>(P1)</b></li> <li>• CPx: Zle zobrazené meno účastníka v konferencii - <b>(P0)</b></li> <li>• CPx: Nie je možné napárovať iPhone 7 - <b>(P1)</b></li> </ul>

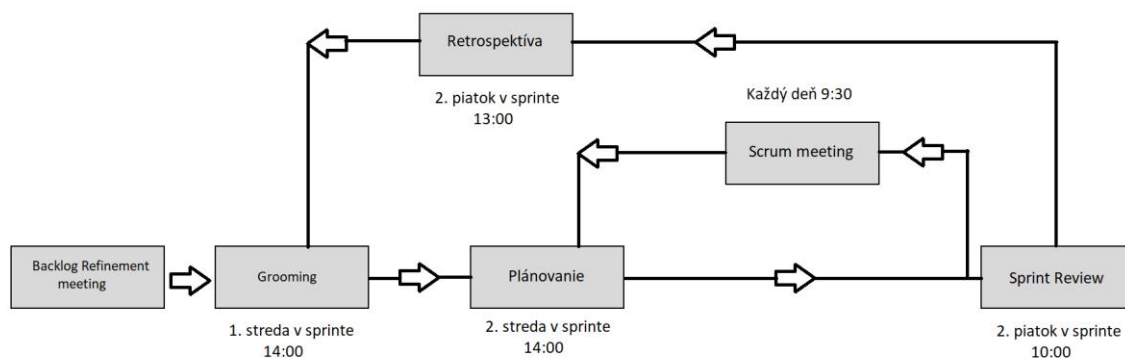
Z nášho pôvodného naplánovaného Sprintu 1 by bol ideálny stav počas Sprint Review Meetingu taký, ako je zobrazený v nasledujúcej tabuľke. To znamená stĺpec „To Do“ by bol prázdny a v stĺpci „Done“ by sa nachádzali všetky úlohy, ktoré boli na začiatku v Sprint Backlogu. Všetky User Stories, ktoré sa nachádzajú v „Done“ by mali byť prezentované zákazníkovi.

### 3.2.10 Grafický návrh procesu vývoja

Na začiatku tejto kapitoly boli navrhnuté role a ich úlohy podľa princípov Scrumu a potrieb vývojových tímov spoločnosti. Na základe toho boli zložené jednotlivé Scrum tímy.

Dôležitou súčasťou Scrumu sú meetingy, bez ktorých by Scrum nebolo možné zaviesť. Priebeh jednotlivých meetingov a ich podrobná náplň bola navrhnutá v predošlých podkapitolách. Tak isto boli spomenuté jednotlivé body, na ktoré bude treba pri zavádzaní Scrumu dať pozor alebo im venovať väčšiu pozornosť. Pri návrhu bolo potrebné brať do úvahy produkt, ktorý spoločnosť vyvíja a tak isto zákazníkov, s ktorými spoločnosť spolupracuje. Niektoré meetingy a role ľudí bolo potrebné konkrétnejšie prispôbiť firemným potrebám.

Pri definovaní meetingov bolo viackrát spomenuté, že väčšinou je na samotnom tíme kedy si nastaví čas každého meetingu. Ale aj pri stanovení termínu meetingov je potrebné zachovať isté pravidlá. Na obrázku je zobrazený konkrétny príklad návrhu, ako by mohol vyzeráť priebeh jedného Sprintu so zoznamom všetkých meetingov, s konkrétnymi časmi pre Scrum Tím vo firme.



**Obrázok 23: Návrh Sprintu s konkrétnym plánom meetingov** (Zdroj: vlastné spracovanie podľa [12])

### **3.3 Nástroje na riadenie Scrumu**

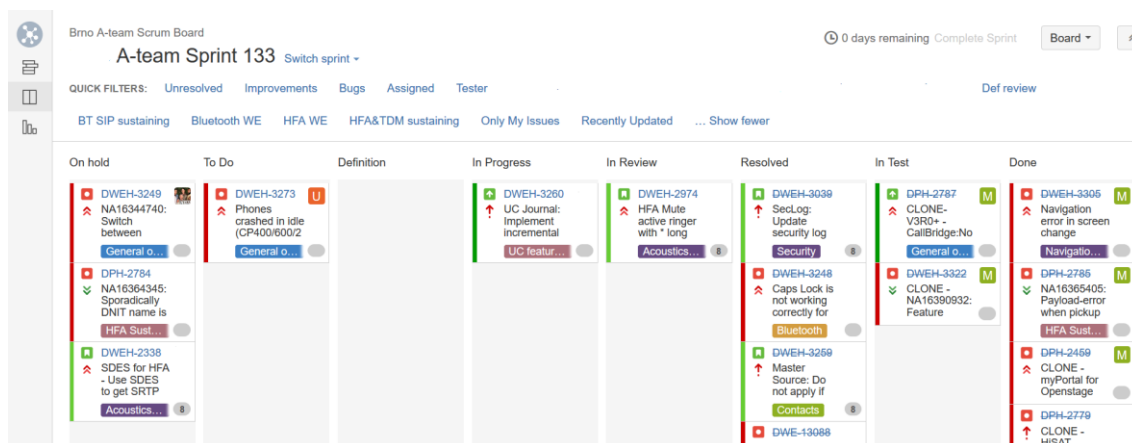
Spoločnosť doteraz používala nástroje na riadenie procesov, ktoré boli staršie a nie tak prehľadné. Konkrétne to bol interný nástroj spoločnosti, ktorý by však nebol vhodný po zavedení Scrumu a jeho riadenie. Preto je potrebné zaviesť nové nástroje, ktoré budú vhodné na nastavenie a riadenie procesov podľa agilných princípov.

#### **3.3.1 Jira Software**

Jira je jeden z najvýznamnejších software, ktorý slúži na riadenie vývoja projektov. Podporuje a uľahčuje proces riadenia projektov a jednotlivých požiadaviek v rámci projektu. Slúži na prehľadnejšie a efektívnejšie sledovanie a evidenciu problémov a úloh v projektoch. Výhodou Jira Software je, že je použiteľný ako nástroj na agilné riadenie, čo znamená, že podporuje rôzne nadstavby, napríklad pre Scrum, Kanban alebo aj špecifickejšie nastavenia riadenia procesu vývoja. V rámci zavedenia Scrum vo firme, čo je hlavnou témou tejto diplomovej práce, by Jira Software bol vhodným nástrojom pre riadenie procesov vo firme, nakoľko ponúka aj nadstavbu pre Scrum s prípadnými špecifickejšími nastaveniami, ktoré by spoločnosť potrebovala.

#### **3.3.2 Scrum Tabula**

Na základe jednotlivých projektov vo firme a možností, ktoré Jira Software ponúka bude navrhnutá Scrum Tabula, ktorá nebude obsahovať len tri stĺpce To Do, In Progress, Done ale budú navrhnuté ďalšie stavy, ktoré budú pridané k trom základným.



Obrázok 24: Scrum Board (Zdroj: vlastné spracovanie)

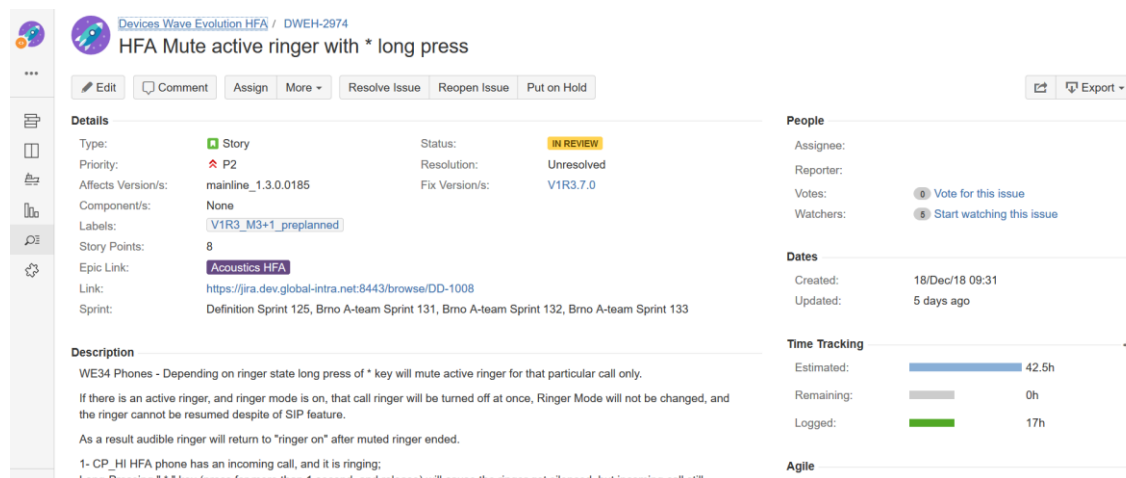
Podľa obrázku vyššie bude navrhnutá Scrum Tabuľa kde bude definovaných 6 stavov, v ktorých sa úlohy v podobe Jira tiketov môžu nachádzať. Jednotlivé stavy budú definované takto:

- On Hold – úlohy sa budú presúvať do stavu On Hold v prípade, keď je člen tímu blokovaný niekým iným, alebo čaká na odpoveď od niekoho iného a pri riešení úlohy sa bez toho nevie posunúť ďalej.
- To Do – úlohy v stave To Do sa nachádzajú väčšinou na začiatku Sprintu, keď sú priradené z Produktového Backlogu do Sprint Backlogu. Členovia tímu si vyberajú z týchto úloh a začínajú na nich pracovať.
- Definition – tento stav je vyhradený pre ľudí, ktorí pracujú v pozícii Product Owner alebo Product Owner Proxy. Sú to úlohy, kde sa pracuje na definovaní jednotlivých User Stories. Tento stav je určený len pre Product Ownerov a Product Owner Proxy, aby ostatní členovia videli na akých úlohách pracujú a aké novinky sa dajú očakávať v Produktovom Backlogu. Úlohy z tohto stavu sa nepresúvajú v rámci Scrum Tabule.
- In Progress – každý člen tímu, ktorý začne pracovať na nejakej úlohe, presúva daný tiket do stavu In Progress pokiaľ nie je hotový.
- In Review – ak vývojár dokončí implementáciu, tiket presúva do stavu In Review, kde čaká na schválenie vlastnej implementácie inými členmi tímu. Bez vykonania review nie je možné tiket posunúť ďalej.
- Done – v tomto stave je úloha zintegrovaná do existujúceho software a z pohľadu vývojára dokončená.

- In Test – pred tým ako je tiket uzavretý ako dokončený, tester alebo iný člen tímu testuje implementovanú úlohu a scenáre, ktoré sú z pohľadu implementácie kritické.
- Resolved – posledný stav Scrum Tabule, ktorý znamená, že úloha bola dokončená. V tomto stave je možné nastaviť aj to, akým spôsobom bola táto úloha dokončená v rámci položky „Resolution“.

### 3.3.3 Úloha v Jira Software

Tiket v Jire predstavuje úlohu, na ktorej členovia tímu pracujú a posúvajú ju cez všetky stavy od To Do až po Resolved. V rámci Jiry je možné nastaviť niektoré veci podľa projektových požiadaviek a stále sa držať princípov Scrumu tak, ako to bolo vidieť v definovaní jednotlivých stavov. Dôvodom návrhu na použitie Jiry je aj popis jednotlivých úloh, ktoré sú v Jire vytvorené. Návrh, ako by úloha vytvorená v Jire mohla vyzeráť aj s popisom je zobrazený na obrázku nižšie.



Obrázok 25: Vzor tiketu v Jira Software (Zdroj: vlastné spracovanie)

Po vytvorení úlohy v Jire bude možné nastaviť jednotlivé parametre pre danú úlohu, ktorými sú napríklad:

- Type – určenie typu úlohy, takže môže sa jednať o User Story, Bug, Improvement, Task, Spike alebo Epic.
- Priority – určenie priority.

- Affects Version – bude sa určovať hlavne pri chybe, kde sa bude zadávať verzia, na ktorej bola chyba objavená.
- Component.
- Label – ak má úloha špecifický Label.
- Environment – v prípade našej spoločnosti sa sem bude zadávať model telefónu alebo typ ústredne.
- Classification – ak má úloha nejakú bližšiu klasifikáciu, tak je možné ju pridať sem.
- Epic Link – nastavenie tohto parametru, pokiaľ úloha patrí pod nejaký väčší epic, používa sa napríklad pri filtrovaní úloh v Jire.
- Sprint – Sprint, do ktorého je úloha priradená.
- Status – Stav, v ktorom sa úloha nachádza (To Do, In Progress,..).
- Resolution – tento parameter sa nastavuje pri zatváraní úlohy.
- Fix Version – verzia, v ktorej bude zmena integrovaná.

Okrem týchto parametrov je možné v rámci Jira úlohy pridať jej popis a tak isto ju komentovať, prípadne pridávať ďalšie prílohy, ktoré s úlohou súvisia alebo jednotlivé úlohy navzájom prepájať. Jira Software obsahuje pomerne veľké množstvo možností, ktoré ponúka a je vhodný pri zavedení Scrumu vo firme.

### 3.4 Časová analýza

Na časový odhad prevedenia zmeny z aktuálneho procesu riadenia vývoja na Scrum bude použitá jedna z metód časovej analýzy. Obsahom tejto podkapitoly bude odhad jednotlivých činností a tak isto identifikovanie činností, ktoré by sa nemali oneskoriť z pohľadu celého projektu. Na začiatku analýzy budú identifikované jednotlivé činnosti, ktoré budú túto zmenu sprevádzať.

Činnosti potrebné na vykonanie zmeny:

- A. Analýza transformácie na agilný prístup za prítomnosti odborníkov.
- B. Nový skúsený Scrum Master (aspoň traja).
- C. Školenie pre top management.
- D. Školenia softskills pre budúcich Scrum Mastrov.
- E. Oboznámiť zákazníkov o zmene.



- F. Prispôsobenie Jira software na Scrum.
- G. Reorganizácia tímov.
- H. Vybrať vhodné nástroje.
- I. Školenia Scrum princípov pre vývojové tímy.
- J. Nastavenie rolí v tímoch.
- K. Nastavenie pravidiel v tímoch.
- L. Pripraviť backlog tikety.
- M. Určenie priorít Product Ownerom v rámci vývoja.
- N. Vytvoriť kalendár meetingov.
- O. Plánovanie školení pre vývojárov.
- P. Dokumentácia zmeny.

### **3.4.1 PERT analýza**

Nakoľko sa takáto zmena vo firme robí po prvýkrát, tak je ťažké odhadnúť jej časové údaje. Preto k tomu využijeme metódu sieťovej analýzy PERT. Táto metóda pozostáva z trojbodového časového odhadu trvania jednotlivých činností (pesimistický, optimistický a najpravdepodobnejší odhad ) a zo skúseností z minulosti. To je hlavný rozdiel od metódy CPM, ktorá predpokladá jeden časový údaj. V nasledujúcej tabuľke sú uvedené údaje, pomocou ktorých budeme vytvárať sieťovú analýzu PERT. Jednotlivé činnosti majú určené svojich predchodcov v nasledujúcej tabuľke.

**Tabuľka 14: Činnosti zmeny s určením predchodcov (Zdroj: vlastné spracovanie)**

Činnosť	Popis činnosti	Predchodca	y <sub>ij</sub> (dní)
<b>A</b>	Analýza transformácie na agilný prístup za prítomnosti odborníkov	-	5
<b>B</b>	Nový skúsený Scrum Master (aspoň dvaja)	A	12
<b>C</b>	Školenie pre top management	B	5
<b>D</b>	Školenia softskills pre Scrum Mastrov	G	7
<b>E</b>	Oboznámiť zákazníkov o zmene	C	15
<b>F</b>	Prispôsobenie Jira software na Scrum	A	14
<b>G</b>	Reorganizácia tímov	B	5
<b>H</b>	Vybrať vhodné nástroje	F	8
<b>I</b>	Školenia Scrum princípov pre vývojové tímy	G	15
<b>J</b>	Nastavenie rolí v tímoch	D,I	5
<b>K</b>	Nastavenie pravidiel v tímoch	J	7
<b>L</b>	Pripraviť backlog tiketov	C	14
<b>M</b>	Určenie priorít Product Ownerom v rámci vývoja	L	4
<b>N</b>	Vytvoriť kalendár meetingov	G	5
<b>O</b>	Plánovanie školení pre vývojárov	N	14
<b>P</b>	Dokumentácia zmeny	E,M,K,O,H	15

Po identifikovaní predchodcov každej jednej činnosti je potrebné stanoviť časové odhady. Činnosti potrebné na vykonanie zmeny s jednotlivými časovými odhadmi sú uvedené nižšie.

**Tabuľka 15: Určenie trvania činnosti, rozptylu, smerodajnej odchýlky (Zdroj: vlastné spracovanie)**

Činnosť	$a_{ij}$	$m_{ij}$	$b_{ij}$	$y_{ij}$	$\sigma^2 y_{ij}$	$\sigma y_{ij}$
<b>A</b>	3	5	7	5	0,44	0,66
<b>B</b>	7	10	20	12	4,694	2,16
<b>C</b>	3	5	7	5	0,44	0,66
<b>D</b>	4	7	10	7	1	1
<b>E</b>	10	15	20	15	2,77	1,66
<b>F</b>	7	14	21	14	5,44	2,33
<b>G</b>	3	5	7	5	0,44	0,66
<b>H</b>	5	7	10	8	0,694	0,83
<b>I</b>	10	15	20	15	2,77	1,66
<b>J</b>	3	5	7	5	0,44	0,66
<b>K</b>	5	7	9	7	0,44	0,66
<b>L</b>	7	14	21	14	5,44	2,33
<b>M</b>	2	4	6	4	0,44	0,66
<b>N</b>	3	5	7	5	0,44	0,66
<b>O</b>	7	14	21	14	5,44	2,33
<b>P</b>	10	15	20	15	2,77	1,66

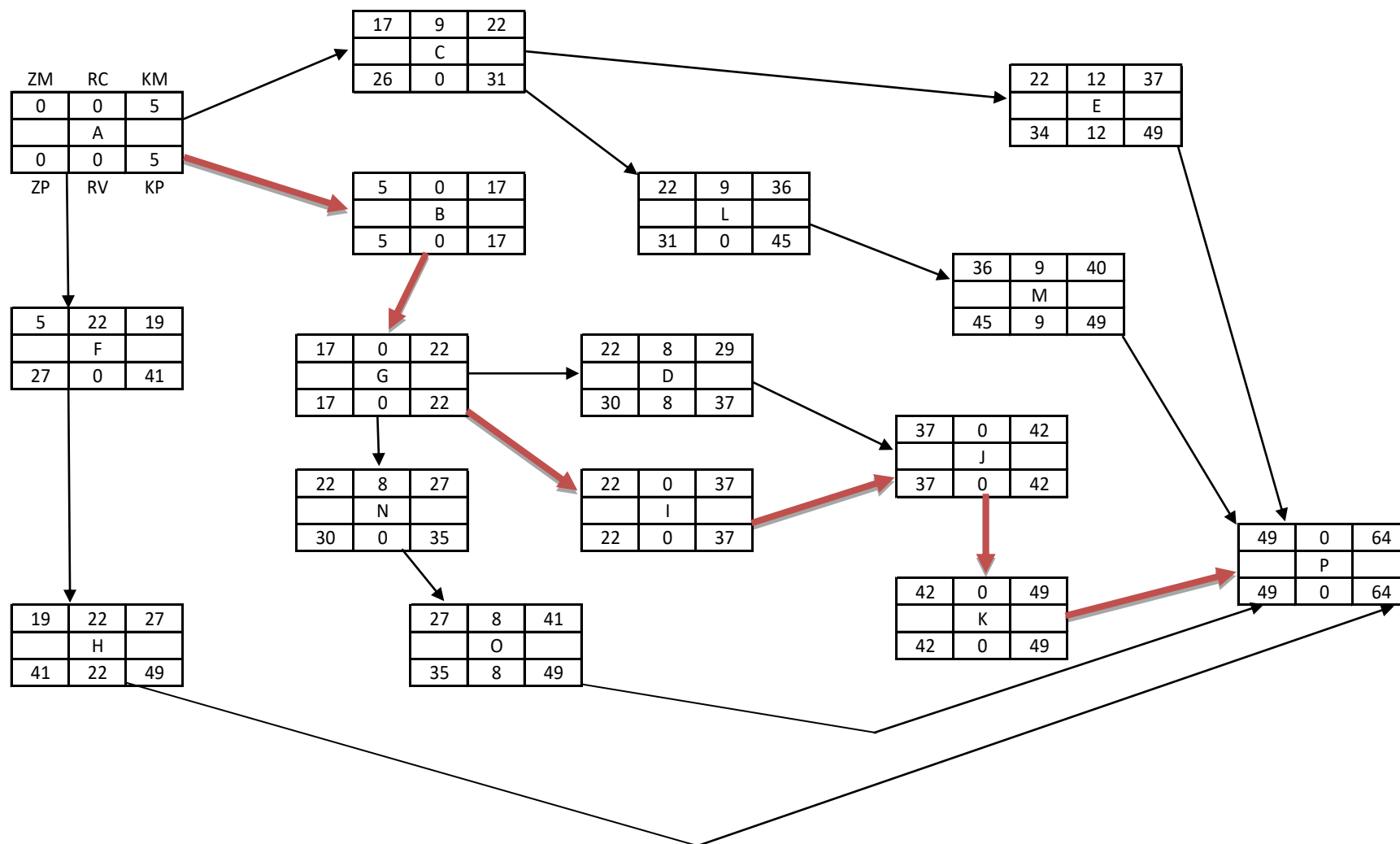
Uvedená tabuľka zachytáva nasledujúce doby trvania ktoré sú uvedené v jednotlivých stĺpcoch:

$a_{ij}$  – optimistický odhad doby trvania

$m_{ij}$  – najpravdepodobnejší odhad doby trvania

$b_{ij}$  – pesimistický odhad doby trvania

$y_{ij}$  – očakávaná doba trvania



**Graf 1: Časový graf PERT analýzy (Zdroj: vlastné spracovanie)**

Na predchádzajúcom obrázku je znázornený sieťový graf PERT analýzy, ktorý vychádza z tabuľky vyššie. Prehľadne sú v ňom znázornené jednotlivé činnosti, ktoré sú nevyhnutné pre realizáciu projektu.

Z grafu vyplýva, že projekt bude ukončený za 64 dní. Kritická cesta je daná činnosťami: A, B, G, I, J, K, P. Tieto činnosti sa nemôžu oneskoriť.

### **3.4.2 Analýza rizík**

Táto kapitola bude venovaná na začiatku hlavne identifikácií rizík, ktoré môžu nastať pri zmene, ktorú plánujeme vo firme zaviesť, a ktorým by sme mali venovať pozornosť.

#### **Kvantifikácia rizík .**

V tabuľke nižšie budú uvedené možné riziká, ktoré budú brané do úvahy. Pri týchto rizikách sú uvedené možnosti výskytu a tak isto aj dopady rizík na zmenu procesu vývoja vo firme.

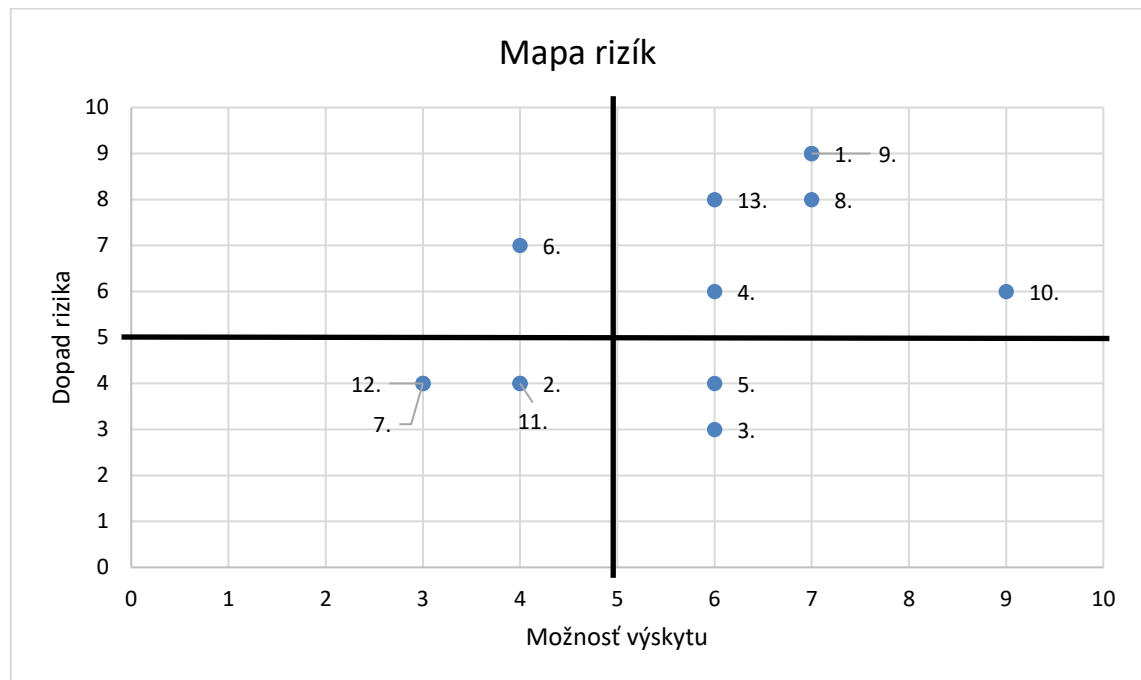
**Tabuľka 16: Určenie rizík a ich kvantifikácia** (Zdroj: vlastné spracovanie)

<b>P.č.</b>	<b>Riziko</b>	<b>Možnosť výskytu (1-10)</b>	<b>Dopad rizika (1-10)</b>	<b>Hodnota rizika</b>
<b>1.</b>	Nedostatočná podpora zo strany managementu	7	9	63
<b>2.</b>	Nadmerný počet meetingov	4	4	16
<b>3.</b>	Zlé priradenie kompetencií v tímoch	6	3	18
<b>4.</b>	Spomalenie procesu vývoja	6	6	36
<b>5.</b>	Dlhá doba zaučenia	6	4	24
<b>6.</b>	Zníženie produktivity	4	7	28
<b>7.</b>	Nezáujem tímu	3	4	12
<b>8.</b>	Rozdielna funkcionálnosť produktu	7	8	56
<b>9.</b>	Nezáujem zákazníka počas vývoja	7	9	63
<b>10.</b>	Nepresné zadania od zákazníka	9	6	54
<b>11.</b>	Nedostatočný počet Scrum Mastrov	4	4	16
<b>12.</b>	Zhoršenie komunikácie	3	4	12
<b>13.</b>	Nepochopenie princípu a výhod zo strany zákazníka	6	8	48

Na analýzu rizík bola použitá skórovacia metóda, na základe ktorej bola vytvorená predošlá tabuľka. Jednotlivé hodnoty boli nanosené na stupnici možnosť výskytu

a dopad na riziko, pričom každé zvlášť na stupnici od 1 do 10. Jednotlivé riziká sa vyhodnocujú na základe toho v akom kvadrante sa nachádzajú.

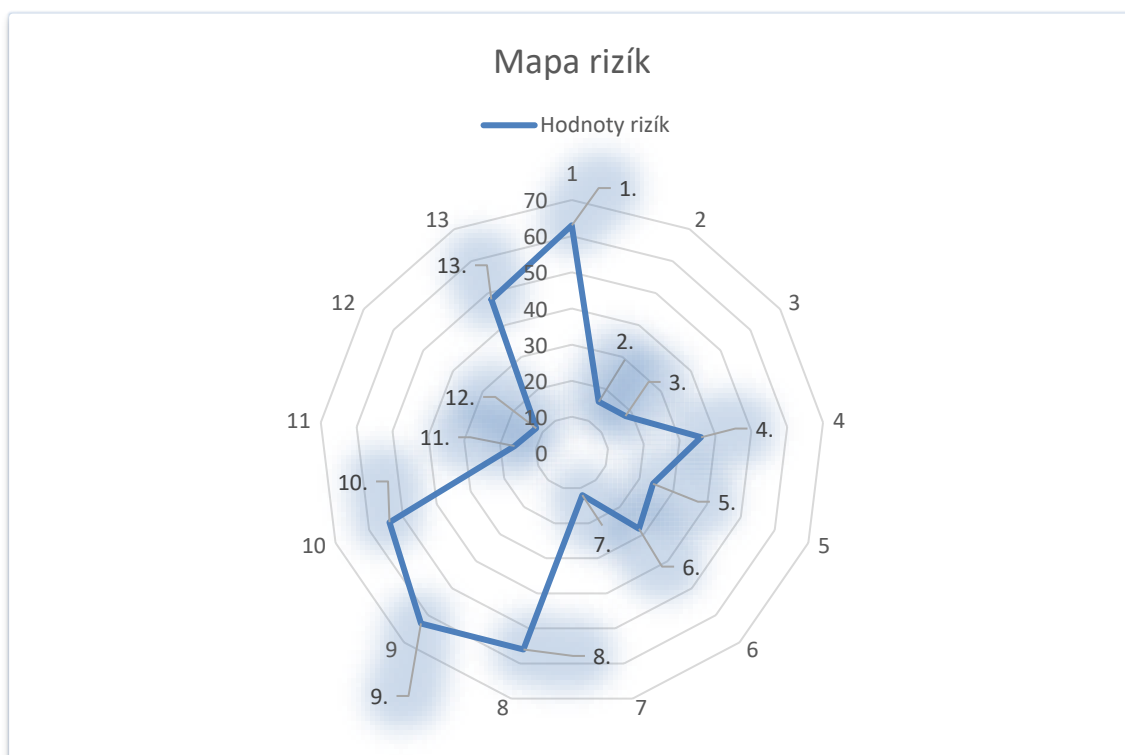
- Vľavo dole - bezvýznamné riziká
- Vpravo dole - bežné riziká
- Vľavo hore - významné riziká
- Vpravo hore – kritické riziká



**Graf 2: Mapa rizík podľa kvadrantov** (Zdroj: vlastné spracovanie)

Na základe tohto grafu boli zistené riziká, na ktoré by sa mala firma zamerať. Sú to riziká, ktoré sa nachádzajú v kvadrante s významnými a kritickými rizikami. Hlavne tie kritické sú pre výsledok projektu dôležité.

Toto rozloženie rizík je zobrazené aj na ďalšom pavučinovom grafe, kde je vidieť ako sú jednotlivé riziká rozložené na základe ich hodnôt.



**Graf 3: Riziká v pavučinovom grafe (Zdroj: vlastné spracovanie)**

Medzi najväčšie riziká patria činnosti úzko súvisiace so zákazníkom, ako je nepresnosť zadania a nezáujem zákazníka počas procesu vývoja. To sú presne riziká, na ktoré je potrebné dávať najväčší pozor pri zavádzaní Scrumu. Výsledkom toho je často rozdielna funkcionálna produkt, ktorá má často negatívny vplyv na ďalší vývoj. Z grafov, ktoré boli predstavené a celej analýzy vyplýva, že jednoznačne najväčším rizikom je malá podpora zo strany vrcholového manažmentu. Toto riziko je veľmi podstatné vnímať, a ak sa začne prejavovať, znamená to, že proces riadenia fungovať nebude a budú časom nastávať komplikácie.

### 3.5 Náklady na zavedenie zmeny vo firme

Na zavedenie prezentovaných zmien vo firme bude potrebné investovať do školení zamestnancov na rôznych pozíciách. Tieto školenia sú potrebné na to, aby táto transformácia prebehla čo najrýchlejšie, bez väčších problémov a hlavne, aby boli princípy Scrumu zavedené do firmy správne a efektívne.



### 3.5.1 Náklady na zavedenie Scrumu

Školenia bude potrebné rozdeliť na rôzne skupiny zamestnancov. Ako bolo už spomenuté, je pravdepodobné prijatie nových zamestnancov na pozície, ako sú Scrum Master, Product Owner a Product Owner Proxy. Ale firma disponuje kvalitnými zamestnancami, ktorí by mohli zastávať tieto nové role a vedeli by sa novým princípom prispôbiť. V každom prípade je potrebné im poskytnúť kvalitné školenia.

Školenia by sa dali rozdeliť na dve hlavné skupiny. V jednej skupine by sa nachádzali ľudia, ktorí by absolvovali školenia pre role Scrum Master a Product Owner a v druhej skupine by boli samotní členovia vývojových tímov.

Pre členov vývojového tímu bude vyčlenený 2-denný intenzívny kurz princípov Scrumu, ktorý budú absolvovať vždy všetci členovia z dvoch tímov naraz. V tabuľke sú uvedené ceny kurzov na jedného človeka.

**Tabuľka 17: Školenie členov tímu** (Zdroj: vlastné spracovanie)

Typ školenia	Počet vývojárov	Cena na osobu	Celková suma
<b>Základné Princípy Scrumu</b>	50	150 €	7500 €

Druhá skupina ľudí bude absolvovať školenia na pozície Scrum Mastrov a Product Ownerov, ktorých obsah sa bude líšiť od kurzov, ktoré absolvujú členovia vývojových tímov. Predpokladá sa zaškolenie vlastných ľudí z firmy, ktorí budú vybraní na tieto pozície a zaškolenie dvoch nových zamestnancov, ktorých firma tak isto pošle na školenia. V prípade, ak by noví zamestnanci už tieto školenia mali absolvované, tak by ich nebolo potrebné posilať na školenia ešte raz. V tomto prípade sa bude jednať o 3-dňový intenzívny kurz, ktorý by mal byť zakončený certifikátom. Tabuľka cien a počet ľudí je uvedený v tabuľke nižšie.

**Tabuľka 18: Školenie Scrum Mastrov a Product Ownerov (Zdroj: vlastné spracovanie)**

<b>Typ školenia</b>	<b>Počet aktuálnych zamestnancov</b>	<b>Počet nových zamestnancov</b>	<b>Cena na osobu</b>	<b>Celková suma</b>
<b>Scrum Master</b>	1	3	650 €	2600 €
<b>Product Owner Proxy</b>	3	1	650 €	2600 €
<b>Product Owner</b>	3	1	650 €	2600 €
<b>Certifikácie</b>	7	5	100 €	1200 €

V druhej tabuľke sú uvedené sumy pre kurz zamestnancov na pozícii Scrum Master a Product Owner. Jedná sa o 3-dňový intenzívny kurz, ktorý by mal pripraviť zamestnancov, ktorí budú zastávať tieto role tak, aby boli schopní plniť úlohy popísané v rámci návrhu v predošlých podkapitolách. Tento kurz bude zakončený certifikáciou, ktorú by mal absolvovať každý prítomný člen.

### **3.5.2 Náklady na zavedenie Jira Software**

Zavedenie Jira Software by malo byť kľúčové pri nasadení princípov Scrumu vo firme. Tento software by mal výrazne sprehľadniť a zefektívniť riadenie vývojového procesu v spoločnosti. Návrh na zavedenie Jira Software by mal byť najskôr v skúšobnej dobe, a ak by sa osvedčil, tak by bol navrhnutý kontrakt na dlhšie obdobie. V tabuľke sú uvedené počty a ceny licencií na tento software, ktoré budú potrebné.

**Tabuľka 19: Jira Software v skúšobnej verzii** (Zdroj: vlastné spracovanie)

	Počet licencií	Cena (cena za mesiac)	Celková suma
<b>Jira Software</b>	60	10 €	600 €

V prípade, ak by sa Jira osvedčila, bol by podpísaný kontrakt na ročné obdobie, kde by bola pre firmu ponúknutá výhodnejšia zmluva. Tabuľky zvýhodnených cien sú uvedené nižšie.

**Tabuľka 20: Jira Software ročná licencia** (Zdroj: vlastné spracovanie)

	Počet licencií	Cena (cena za mesiac)	Celková suma
<b>Jira Software (kontrakt na rok)</b>	60	7 €	420 €

K transformácii zo starého procesu riadenia na Scrum by mali výrazne dopomôcť odborníci na Scrum, ktorí by mali byť prítomní vo firme počas prvého mesiaca zavedenia zmeny. Mali by mať na starosti kontrolovanie priebehu zmeny a riešiť prípadné komplikácie jednotlivých zamestnancov.

**Tabuľka 21: Náklady na špecialistov na Scrum** (Zdroj: vlastné spracovanie)

	Počet	Plat (na mesiac)	Celková suma
<b>Scrum špecialista</b>	2	2000 €	4000€

Na zavedenie zmeny bude potrebné investovať väčší obnos peňazí, ale v budúcnosti by sa táto zmena mohla prejavovať pozitívne. V tabuľke sú uvedené jednotlivé ceny ktoré by mali byť investované do procesu zmeny.

**Tabuľka 22: Celkové náklady na zavedenie zmeny (Zdroj: vlastné spracovanie)**

<b>Činnosť</b>	<b>Suma</b>
<b>Kurz členov vývojových tímov</b>	7 500 €
<b>Kurz a certifikácia (Product Owner + Product Owner Proxy + Scrum Master)</b>	9000 €
<b>Troja noví Scrum Mastri (mesačný plat na jedného)</b>	1300 €
<b>Nový Product Owner Proxy (mesačný plat)</b>	1800 €
<b>Nový Product Owner (mesačný plat)</b>	2000 €
<b>Jira Software (na mesiac)</b>	420 €
<b>Náklady na Scrum Špecialistov</b>	4000 €
<b>Celková suma</b>	26 020 €

Nakoľko firma bude naberat' troch nových Scrum Mastrov, jedného človeka na pozíciu Product Owner a Product Owner Proxy, mesačné mzdy narastú o 7 700 €. Zmena procesu riadenia a nasadenie Scrumu nebude mať vplyv na platy ostatných zamestnancov.

Ak budú jednotlivé zmeny urobené podľa prezentovaného návrhu, tak by absolvovanie kurzov s certifikátmi pre Product Ownerov a Scrum Mastrov a zavedenie Jira Software

na prvý mesiac malo vyjsť 26 020 €. Tieto sumy nemusia byť presné a môžu sa meniť podľa podpísaných zmlúv zo spoločnosťami, ktoré tieto služby ponúkajú.

### **3.6 Prínos návrhov**

Zmena prístupu riadenia vývoja vo firme je nevyhnutným krokom, ktorý by mal zlepšiť aktuálny stav vo firme a na jednotlivých projektoch. Proces riadenia podľa starej metodiky, kde bol využívaný najmä vodopádový model vývoja sa na viacerých projektoch v plnej miere neosvedčil, a niektoré projekty zlyhávali najmä kvôli zásadným problémom týchto tradičných metód. Veľakrát viazla komunikácia medzi zákazníkom a vývojovými tímami alebo výsledná funkcionálnosť produktu sa líšila od takej, ktorú si zákazník na začiatku predstavoval. Podobne si bolo nutné uvedomiť, že v rámci starého prístupu tímy nie sú schopné rýchlo reagovať na zmeny požiadaviek pružne a veľakrát kvôli tomu musí vývoj začínať odznova.

Návrh zmeny prístupu riadenia na agilný za pomoci metodiky Scrum by mal byť vhodným riešením vyššie spomenutých problémov, nakoľko tento návrh by mal zákazníka začleniť do procesu vývoja a tak by sa firma mala vyhnúť problémom s chýbajúcou komunikáciou so zákazníkom a flexibilnejšie reagovať na zmenu požiadaviek. Zahŕnutie požiadaviek do jednotlivých Sprintov zakončených meetingom tzv. Sprint Review by malo byť efektívnejšie a najmä zákazník by mal možnosť vidieť naimplementované zmeny na vytvorenom prototype, a popri prípade konzultovať tieto zmeny priamo s členmi Scrum tímu pravidelne na týchto meetingoch. V prípade zmeny požiadaviek, by tím mohol zaradiť dodatočné zmeny do ďalších Sprintov a pracovať na nich.

Zavedenie Scrumu by mohlo zlepšiť aj komunikáciu a atmosféru medzi jednotlivými členmi tímov. To by malo dopomôcť k motivovaniu jednotlivých zamestnancov, aby do produktu prispeli vlastnými návrhmi a zlepšeniami.

Napriek tomu, že firma bude musieť vynaložiť nemalé finančné prostriedky na transformáciu starého procesu hlavne v podobe školení a kurzov je to dobrým prísľubom do budúcnosti, ktorý môže spoločnosti ponúknuť aj konkurenčné výhody na trhu pri získavaní nových projektov. Súčasťou návrhu zmeny by malo byť aj zavedenie Jira Software, ktorý by mal fungovať ako podporný nástroj pri riadení vývoja pomocou metodiky Scrum.

## ZÁVER

Cieľom tejto diplomovej práce bolo zavedenie a zefektívnenie procesu vývoja pomocou agilnej metodiky Scrum vo firme iXperta, ktorá sa orientuje na vývoj softwaru VoIP zariadení.

Prvá časť diplomovej práce je orientovaná na vymedzenie a definovanie základných pojmov týkajúcich sa projektového managementu, tradičných metód vývoja softwaru a jednotlivých typov agilných metód. Pomerne veľká časť opisuje základné pojmy, artefakty a meetingy, ktoré sú neoddeliteľnou súčasťou Scrumu.

Dôležitou kapitolou je časť, týkajúca sa analýzy súčasnej situácie vo firme, na ktorú boli použité metódy SLEPT, 7S, Porterova analýza piatich síl a SWOT analýza, pomocou ktorej boli definované silné a slabé stránky a tak isto hrozby a príležitosti firmy. Hlavným problémom sa ukázali byť veci, ktoré úzko súviseli so zákazníkom a jeho začlenením do procesu vývoja. Zmena požiadaviek, alebo nepresné zadanie požiadaviek na začiatku vývoja patrili medzi časté problémy, na ktorých firma strácala či už finančné alebo časové prostriedky. V tejto kapitole je tak isto bližšie popísaný priebeh vývoja vo firme a jeho jednotlivé fázy.

Na základe analytickej časti, z ktorej vyplynulo, že firma potrebuje zmenu, bolo navrhnuté zavedenie agilnej metodiky Scrum v rámci celej firmy. Návrhová časť popisuje zavedenie nových rolí v jednotlivých tímoch, nastavenie základných artefaktov Scrumu podľa potrieb spoločnosti a požiadaviek zákazníka, ktorý zohráva v nasadení Scrumu dôležitú úlohu. Na príklade je predstavené zostavenie Produktového Backlogu s definovaním jednotlivých User Stories s naplánovaním Sprintu 1, na ktorom bude pracovať vývojový tím. Neoddeliteľnou súčasťou Scrumu sú meetingy, ktorých obsah a hlavná úloha sú navrhnuté postupne. Na konci je zobrazený grafický návrh priebehu Sprintu pre Scrum tím, kde je navrhnutý časový plán všetkých meetingov. Výrazným zľahčením by malo byť nasadenie Jira Software, ktoré by malo uľahčiť dodržiavanie princípov Scrumu a zjednodušiť celý proces.

Samotné kapitoly sú orientované na zobrazenie časovej analýzy jednotlivých činností, ktoré sú potrebné na zmenu a tak isto analýze rizík, ktorej výsledkom sú riziká vyplývajúce zo zmeny procesu riadenia.

Výsledkom tejto zmeny by malo byť výrazné začlenenie zákazníka do vývoja, kde by bol hlavnou postavou pri definovaní požiadaviek, ktoré si môže po ich naimplementovaní vyskúšať. Zmena by mala prispieť k zefektívneniu riadenia projektov, ktorá by mala viesť aj k lepším finančným príjmom aj keď si na začiatku bude vyžadovať nezanedbateľné náklady. Nasadenie Scrumu, tak ako je popísané v návrhovej časti, môže byť použiteľné aj pri aplikovaní v iných firmách, ktoré nepôsobia len v oblasti informačných technológií.

## ZOZNAM POUŽITEJ LITERATÚRY

- 1) KERZNER, Harold. *Project management: a systems approach to planning, scheduling, and controlling*. 8th ed. New York: Wiley, 2003, xx, 891 s. ISBN 0-471-22577-0
- 2) *A guide to the project management body of knowledge*. Sixth Edition. Newtown Square: Project Management Institute, 2017, xxix, 756 s. ISBN 978-1-62825-184-5.
- 3) A. SVOZILOVÁ. Projektový management. 2. vyd. Praha: Grada, 2011, 392 s. ISBN 978-80-247-3611-2.
- 4) DOLEŽAL, J. a kol. Projektový management podle IPMA. 1. vyd. Praha: Grada Publishing, 2009. 512 s. ISBN 978-80-247-2848-3.
- 5) DOLEŽAL, Jan. Projektový management: komplexně, prakticky a podle světových standardů. Praha: Grada Publishing, 2016. ISBN 978-80-247-5620-2.
- 6) HAČKAJLOVÁ, Ludmila, Zita PROSTĚJOVSKÁ a Jaroslava TOMÁNKOVÁ. Projektový management. Praha: Vysoká škola ekonomie a managementu, 2013. ISBN 978-80-87839-00-3
- 7) JEŽKOVÁ, Zuzana, Hana KREJČÍ, Branislav LACKO a Jaroslav ŠVEC. *Projektové řízení: jak zvládnout projekty*. Kuřim: Akademické centrum studentských aktivit, 2013, 381 stran : ilustrace (některé barevné). ISBN 978-80-905297-1-7.
- 8) DOLANSKÝ, Václav, Vladimír MĚKOTA a Vladimír NĚMEC. *Projektový management*. Praha: Grada Publishing, 1996, 372 s. ISBN 80-7169-287-5.
- 9) NĚMEC, Vladimír. *Projektový management*. Praha: Grada, 2002, 182 s. : il. ISBN 80-247-0392-0.
- 10) Kočí, R., Křena, B.: *Úvod do softwarového inženýrství*. Studijní opora, Vysoké učení technické v Brně, 2010
- 11) MYSLÍN, Josef. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016, 167 stran : ilustrace. ISBN 978-80-251-4650-7.
- 12) ŠOCHOVÁ, Zuzana a Eduard KUNCE. *Agilní metody řízení projektů*. Brno: Computer Press, 2014, 175 s. : il., portréty. ISBN 978-80-251-4194-6.



- 13) PROCHÁZKA, Jaroslav a Cyril KLIMEŠ. *Provozujte IT jinak: agilní a štíhlý provoz, podpora a údržba informačních systémů a IT služeb*. Praha: Grada, 2011, 288 s. : il., grafy, tab. ISBN 978-80-247-4137-6.
- 14) RAIS, Karel a Radek DOSKOČIL. *Risk management: studijní text pro kombinovanou formu studia*. Brno: Akademické nakladatelství CERM, 2007, 152 s. : il. ISBN 978-80-214-3510-0.
- 15) RAIS, Karel a Radek DOSKOČIL. *Operační a systémová analýza I: studijní text pro prezenční a kombinovanou formu studia*. Brno: Akademické nakladatelství CERM, 2011, 125 s. : il., grafy, tab. ISBN 978-80-214-4364-8.
- 16) HNILICA, Jiří a Jiří FOTR. *Aplikovaná analýza rizika ve finančním managementu a investičním rozhodování*. Praha: Grada, 2009, 262 s. : il. ISBN 978-80-247-2560-4.
- 17) SMEJKAL, Vladimír a Karel RAIS. *Řízení rizik*. Praha: Grada, 2003, 270 s. : il. ISBN 80-247-0198-7.
- 18) SUTHERLAND, J. a K. SCHWABER. Scrumguide. [online] 2013.  
[cit. 2019-04-02]. Dostupné z:  
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- 19) KEŘKOVSKÝ, Miloslav a Oldřich VYKYPĚL. *Strategické řízení: teorie pro praxi*. 2. vyd. Praha: C.H. Beck, 2006, xiv, 206 s. : il., grafy, tab. ISBN 80-7179-453-8.
- 20) HRON, Jan, Jan DOHNAL a Ivana TICHÁ. *Strategické řízení*. Vyd. 2. přeprac. Praha: Credit, 1998. ISBN 80-213-0429-4.
- 21) OpenScape Desk Phone CP600. *Unify atos collaboration solutions* [online]. [cit. 2019-05-07].  
Dostupné z: <https://unify.com/en/solutions/phones-and-clients/desktop-phones>
- 22) Logo Ixperta. *Ixperta* [online]. [cit. 2019-05-07].  
Dostupné z: <https://www.ixperta.com/>

## ZOZNAM TABULIEK

<b>Tabuľka 1: SWOT analýza</b> (Zdroj: vlastné spracovanie).....	22
<b>Tabuľka 2: Uzol sieťového grafu PERT metódy</b> (Zdroj: vlastné spracovanie).....	27
<b>Tabuľka 3: Pigs and Chickens</b> (Zdroj: vlastné spracovanie podľa [12]) .....	48
<b>Tabuľka 4: Nastavenie dĺžky Sprintu</b> (Zdroj: vlastné spracovanie podľa [11]).....	53
<b>Tabuľka 5: Základné informácie o firme iXperta s.r.o</b> (Zdroj: vlastné spracovanie) .....	57
<b>Tabuľka 6: Podrobný popis jednotlivých fáz a podfáz</b> (Zdroj: vlastné spracovanie)67	
<b>Tabuľka 7: SWOT analýza</b> (Zdroj: vlastné spracovanie).....	70
<b>Tabuľka 8: Návrh pozície Scrum Master</b> (Zdroj: vlastné spracovanie).....	75
<b>Tabuľka 9 : Návrh pozície Product Owner</b> (Zdroj: vlastné spracovanie).....	76
<b>Tabuľka 10: Návrh pozície Product Owner Proxy</b> (Zdroj: vlastné spracovanie) .....	77
<b>Tabuľka 11: Produktový Backlog s určenými prioritami</b> (Zdroj: vlastné spracovanie).....	88
<b>Tabuľka 12: Naplánovanie Sprintu 1</b> (Zdroj: vlastné spracovanie ) .....	93
<b>Tabuľka 13: Sprint 1 Review</b> (Zdroj: vlastné spracovanie) .....	99
<b>Tabuľka 14: Činnosti zmeny s určením predchodcov</b> (Zdroj: vlastné spracovanie) .....	106
<b>Tabuľka 15: Určenie trvania činnosti, rozptylu, smerodajnej odchýlky</b> (Zdroj: vlastné spracovanie).....	107
<b>Tabuľka 16: Určenie rizík a ich kvantifikácia</b> (Zdroj: vlastné spracovanie).....	110
<b>Tabuľka 17: Školenie členov tímu</b> (Zdroj: vlastné spracovanie) .....	113
<b>Tabuľka 18: Školenie Scrum Mastrov a Product Ownerov</b> (Zdroj: vlastné spracovanie).....	114
<b>Tabuľka 19: Jira Software v skúšobnej verzii</b> (Zdroj: vlastné spracovanie).....	115
<b>Tabuľka 20: Jira Software ročná licencia</b> (Zdroj: vlastné spracovanie).....	115
<b>Tabuľka 21: Náklady na špecialistov na Scrum</b> (Zdroj: vlastné spracovanie).....	115
<b>Tabuľka 22: Celkové náklady na zavedenie zmeny</b> (Zdroj: vlastné spracovanie) ..	116

## ZOZNAM OBRÁZKOV

<b>Obrázok 1: Trojimperatív projektu (Zdroj: vlastné spracovanie podľa [7])</b>	19
<b>Obrázok 2: Fázy projektu (Zdroj: vlastné spracovanie podľa [8])</b>	20
<b>Obrázok 3: Porterova analýza piatich síl (Zdroj: [14])</b>	23
<b>Obrázok 4: Mapa Rizík - rozdelenie kvadrantov (Zdroj: [7])</b>	32
<b>Obrázok 5: Klasický vs agilný prístup (Zdroj: Vlastné spracovanie)</b>	33
<b>Obrázok 6: Vodopádová metóda (Zdroj: podľa [11])</b>	34
<b>Obrázok 7: Iteratívny model (Zdroj: [10])</b>	35
<b>Obrázok 8: Extrémne programovanie (Zdroj: [12])</b>	42
<b>Obrázok 9: metodika SCRUM Development Process (Zdroj: vlastné spracovanie podľa [12])</b>	46
<b>Obrázok 10: Logo Ixperta (Zdroj: [22])</b>	58
<b>Obrázok 11: Analýza interných faktorov (Zdroj: [14])</b>	60
<b>Obrázok 12: OpenScape Desk Phone CP600 (Zdroj : [21])</b>	61
<b>Obrázok 13: Proces vývoja projektu (Zdroj: Vlastné spracovanie)</b>	66
<b>Obrázok 14: Návrh Scrum Tímu (Zdroj: vlastné spracovanie)</b>	80
<b>Obrázok 15: Product Owner Proxy a Scrum Master a dva vývojové tímy (Zdroj: vlastné spracovanie)</b>	81
<b>Obrázok 16: Product Backlog a Sprint Backlog (Zdroj: vlastné spracovanie podľa [11])</b>	85
<b>Obrázok 17: Konkrétne nastavenie dĺžky Sprintu (Zdroj: vlastné spracovanie podľa [12])</b>	86
<b>Obrázok 18: Planning Poker karty (Zdroj: [12])</b>	90
<b>Obrázok 19: Priority v Produktovom Backlogu (Zdroj: vlastné spracovanie)</b>	92
<b>Obrázok 20: Návrh Scrum tabule (Zdroj: vlastné spracovanie)</b>	95
<b>Obrázok 21: Rozdelenie tabule na retrospektívu (Zdroj: vlastné spracovanie)</b>	97
<b>Obrázok 22: Návrh aplikácie na retrospektívu (Zdroj: vlastné spracovanie)</b>	97
<b>Obrázok 23: Návrh Sprintu s konkrétnym plánom meetingov (Zdroj: vlastné spracovanie podľa [12])</b>	100
<b>Obrázok 24: Scrum Board (Zdroj: vlastné spracovanie)</b>	102

<b>Obrázok 25: Vzor tiketetu v Jira Software (Zdroj: vlastné spracovanie) .....</b>	<b>103</b>
---	------------

## **ZOZNAM GRAFOV**

<b>Graf 1: Časový graf PERT analýzy (Zdroj: vlastné spracovanie) .....</b>	<b>108</b>
<b>Graf 2: Mapa rizík podľa kvadrantov (Zdroj: vlastné spracovanie).....</b>	<b>111</b>
<b>Graf 3: Riziká v pavučinovom grafe (Zdroj: vlastné spracovanie).....</b>	<b>112</b>